Leakage-Aware Dynamic Thermal Management of 3D Memories

LOKESH SIDDHU, RAJESH KEDIA, and PREETI RANJAN PANDA,
Department of Computer Science and Engineering, Indian Institute of Technology Delhi

3D memory systems offer several advantages in terms of area, bandwidth, and energy efficiency. However, thermal issues arising out of higher power densities have limited their widespread use. While prior works have looked at reducing dynamic power through reduced memory accesses, in these memories, both leakage and dynamic power consumption are comparable. Furthermore, as the temperature rises, the leakage power increases, creating a thermal-leakage loop. We study the impact of leakage power on 3D memory temperature and propose turning OFF specific memory channels to meet thermal constraints. Data is migrated to a 2D memory before closing a 3D channel. We introduce an analytical model to assess the 2D memory delay and use the model to guide data migration decisions. The above strategy is referred to as *FastCool* and provides an improvement of 22%, 19%, and 32% on average (up to 57%, 72%, and 82%) in performance, memory energy, and energy-delay product (EDP), respectively, on different workloads consisting of SPEC CPU2006 benchmarks.

We further propose a thermal management strategy named *Energy-Efficient FastCool (EEFC)*, which improves upon FastCool by selecting the channels to be closed by considering temperature, leakage, access rate, and position of various 3D memory channels at runtime. Our experiments demonstrate that EEFC leads to an additional improvement of up to 30%, 30%, and 51% in performance, memory energy, and EDP compared to FastCool. Finally, we analyze the effects of process variations on the efficiency of the proposed FC and EEFC strategies. Variation in the manufacturing process causes changes in the leakage power and temperature profile. Since EEFC considers both while selecting channels for closure, it is more resilient to process variations and achieves a lower application execution time and memory energy compared to FastCool.

CCS Concepts: • **Hardware → Memory and dense storage**; **Dynamic memory**; **3D integrated circuits**; **Temperature simulation and estimation**; **Temperature control**; *Process variations*;

Additional Key Words and Phrases: 3D memories, dynamic thermal management, leakage power, energy efficiency

ACM Reference format:
Lokesh Siddhu, Rajesh Kedia, and Preeti Ranjan Panda. 2020. Leakage-Aware Dynamic Thermal Management of 3D Memories. *ACM Trans. Des. Autom. Electron. Syst.* 26, 2, Article 12 (October 2020), 31 pages.
https://doi.org/10.1145/3419468

Lokesh Siddhu and Rajesh Kedia were supported by Visvesvaraya PhD Scheme, MeitY, Government of India MEITY-PHD-2672 and MEITY-PHD-2671, respectively.
This article is an extension of the paper titled "FastCool: Leakage Aware Dynamic Thermal Management of 3D Memories" published in DATE 2019.
Authors' address: L. Siddhu, R. Kedia, and P. R. Panda, Department of Computer Science and Engineering, Indian Institute of Technology Delhi, Hauz Khas, New Delhi, Delhi, 110016, India; emails: {siddhulokesh, kedia, panda}@cse.iitd.ac.in.

1084-4309/2020/10-ART12 $15.00
https://doi.org/10.1145/3419468

1 INTRODUCTION

To ease performance bottlenecks caused by memory systems, researchers have proposed 3D (stacked 2D) dynamic random access memory (DRAM), which offers a smaller form factor and higher bandwidth. However, 3D memories exhibit a higher power density leading to frequent heating and cooling phases. Power consumption in memories comprises dynamic power and static/leakage power. Static power increases exponentially with temperature creating positive feedback between temperature and leakage [50]. As a significant (~52%) portion of the power consumed by memories is static power [18], it is essential to reduce static power to meet thermal constraints. Obtaining high performance and lower energy consumption under thermal constraints is a challenge in 3D memories and needs research attention [31]. Specifically, reduction of static power has not been addressed in prior works on thermal management of 3D memories [16, 31].

We propose a novel data migration–based dynamic thermal management (DTM) strategy, namely, FastCool (FC) [43], to improve performance and reduce memory energy under thermal constraints for architectures with both 2D and 3D memories (Figure 1). Such architectures utilize the 2D memory to alleviate the high cost per bit and thermal issues of 3D memories [16]. Once the 3D memory heats up, data from 3D memory channels is migrated to the 2D memory and these channels are turned OFF. Migrated data, if required by the processor, is accessed from the 2D memory. When the 3D memory cools down, data is brought back from the 2D memory. We also develop an analytical model to predict the 2D memory delay and use it to perform a cost-benefit analysis to guide migration decisions. We further propose *Energy-Efficient FastCool* (EEFC), a three-pass DTM strategy which improves upon FC by determining the 3D memory channels to be closed at runtime, based on the temperature, leakage, access rate, and position of the channels.

Compared to the state-of-the-art, FC results in an improvement of 22%, 19%, and 32% on average (up to 57%, 72%, and 82%) in performance, energy, and energy-delay product, respectively, on different workloads consisting of SPEC CPU2006 benchmarks. We obtain an additional improvement of up to 30%, 30%, and 51% in performance, energy, and energy-delay product using EEFC. Variation in the manufacturing process causes changes in the leakage power and temperature profile. Since EEFC considers both leakage power and temperature while selecting channels for closure at runtime, it is resilient to process variations which we study in this article through detailed experimentation.

In particular, we make the following key contributions in this article:

(1) We present the first study of the thermal-leakage loop resulting in a system-level dynamic thermal management (DTM) strategy (FC) based on data migration to the 2D memory. We also propose a detailed analytical model to quantify the memory delay, including queuing delays for a system with hybrid 3D and 2D memory.
(2) We present an improved DTM strategy, EEFC which incorporates additional designer insights to improve performance and energy further. We provide a comprehensive evaluation (in terms of execution time and memory energy consumption) of the proposed strategies using 10 different workloads comprising different mixes of SPEC CPU2006 benchmarks.
(3) We present an analysis of the resilience of EEFC toward variability in the manufacturing process.

The rest of this article is organized as follows. We provide the necessary background about 3D memories and motivate the need for leakage power–aware thermal management of 3D memories in Section 2. Section 3 discusses the related prior work. We discuss our proposed strategies, FC and

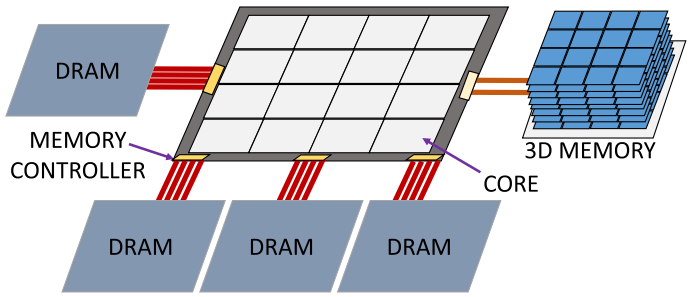


Fig. 1. 16-core processor with off-chip 3D and 2D memories.

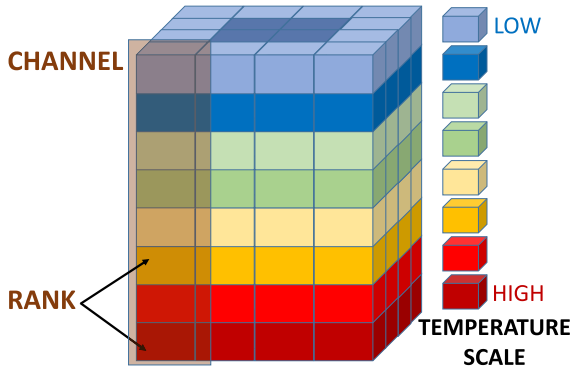


Fig. 2. Sixteen channel 3D memory. Hottest at bottom layer, central ranks.

EEFC, in Section 4. Section 5 discusses our simulation-based experimental setup and evaluation of the proposed strategies on various standard benchmarks. We study the effect of manufacturing process variation on 3D memory thermal management in Section 6. We conclude the article and indicate future directions in Section 7.

2 BACKGROUND AND MOTIVATION

Leakage power forms a significant portion (~52%) of the total power consumed by memories [18], and positive feedback between temperature and leakage leads to the leakage power increasing exponentially with temperature [50]. However, prior works in thermal management of 3D memories have focused on reducing the dynamic power alone [16, 31]. This situation motivates us to explore approaches that also reduce static power during the thermal management of 3D memories. In the following text, we illustrate the need for leakage-aware thermal management by studying the thermal characteristics of 3D memories and quantifying the effect of leakage power on 3D memory temperature.

Thermal Characteristics of 3D Memory Architectures. We discuss the thermal characteristics of Hybrid Memory Cube (HMC), which is one of the industry standard 3D memory architectures [9, 20]. A typical 1 GB HMC consists of eight memory layers (each 128 MB) and one logic layer. The logic layer includes the I/O interface, and control logic. HMC is divided into 16 partitions (Figure 2), and a vertically stacked partition (rank) forms a vault (channel). Other industry-standard architectures such as High Bandwidth Memory (HBM) and Wide IO (WIO) memory have similar vertical channels. A heat spreader and heat sink are placed on top of the 3D memory for cooling. Hence, the upper layers have better heat dissipation capability than the lower layers (Figure 2).

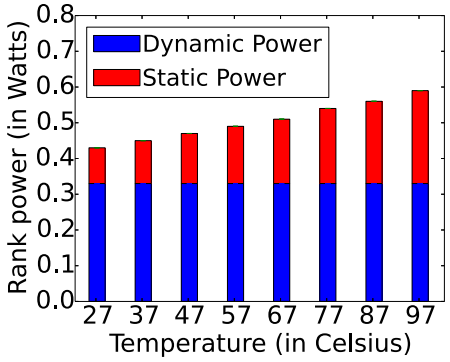


Fig. 3. Rank (8 MB) power vs. temperature for a 3D memory using CACTI-3DD.

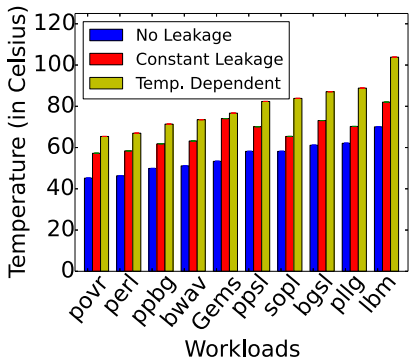


Fig. 4. Effect of leakage power on the 3D memory steady-state temperature.

Within a layer, under uniform power dissipation, the central partitions get heated from all the sides and exhibit the highest temperatures [34].

Effect of Temperature-Leakage Dependence. DRAM manufacturers increase memory capacity by packing more cells per unit area. However, each cell contributes to leakage due to non-idealities in transistors and capacitors causing significant static power dissipation in memories. Furthermore, leakage power increases with rising temperature. We quantify the effect of temperature (Figure 3) on the rank leakage and dynamic power for a 1 GB, 50 nm, 46 mm^2, 128 GBps 3D memory [20] using CACTI-3DD [8]. At higher temperatures, static power contributes significantly to the total power (e.g., 38.8% at 77°C).

We use the temperature-dependent leakage power shown in Figure 3 to study the steady-state temperature of 3D memory for various workloads. Figure 4 compares the steady-state memory temperature when not considering the leakage power (*no leakage*), considering a constant leakage power (obtained for 25°C) independent of temperature (*constant leakage*), and considering the temperature-dependent leakage power (*temp. dependent*) for various workloads (simulation setup and workload details are presented in Section 5). Memory-intensive workloads (toward the right side) induce the highest temperatures. The thermal-leakage loop leads to increased temperatures, with its effect being dominant at higher temperatures. For example, for the *lbm* workload, leakage power increases the temperature by ~25°C. Moreover, a constant leakage model is unable to determine temperature accurately for all workloads. This corroborates the need to consider temperature-dependent leakage power during DTM of 3D memories. It also signifies that the DTM techniques that reduce the dynamic power alone are not sufficient to provide adequate cooling in 3D memories, and DTM strategies should incorporate reduction of leakage power as well.

With this motivation, we present the related prior works and then discuss our proposed thermal-aware system optimization strategy in the subsequent sections.

3 RELATED WORK

Research in thermal-aware design of processors and memories started in the 2000s [3, 4, 11, 13, 16, 17, 21, 22, 23, 24, 30, 34, 37, 38, 45]. Both static [12] and dynamic (runtime) [40, 41] thermal management techniques have been studied. Static methods use design-time temperature profiling to ensure that the hot function units are spread out in the floorplan [12], reducing the maximum temperature. However, they do not scale well to general purpose processing as different applications may show different temperature profiles.

DTM policies periodically monitor the temperature and reduce the access rate of hot functional units using throttling or dynamic voltage/frequency scaling (DVFS). Limiting the performance and energy overhead of DTM schemes is an active area of research.

Several recent works have addressed the thermal management of 3D processors [26, 32, 35]. Liao et al. [26] suggest assigning voltages to cores based on temperature. Voltages are assigned such that cores do not heat up frequently. As part of a DVFS strategy, they also use task scheduling within the DTM policy to balance out power across vertically stacked cores. For a 3D network on chip, Zheng et al. [53] propose a combined approach of adjusting bus frequency and routing to limit the maximum temperature.

While DTM of 3D processors has received considerable attention, there has been limited work addressing thermal issues in 3D memories. Hajkazemi et al. [16] reduce memory latency by distributing pages between 2D and 3D memories (heterogeneous memory architecture) in a ratio that is decided at design time using simulations. Lo et al. [31] discuss a thermal-aware page allocation policy for 3D memories where pages allocated to a hot channel are reallocated to the coldest channel. In both these works [16, 31], DTM is triggered only when there is a request for allocation of a new page. However, accesses to existing pages in the memory can also cause heating and lead to eventual throttling.

For reducing static power in 2D memories, Lu et al. [33] group pages with similar access locality into the same rank and place idle ranks in low power modes. However, none of the prior works in 3D memory thermal management attempt to reduce static power, which extends the duration of the cool-down period. Further, leakage power increases with a rise in temperature and there is a very limited study of the effect of thermal-leakage loop and incorporating its effect in DTM. In other prior works, a constant temperature-independent leakage value is used [34] or the impact of leakage is not mentioned [31]. Incorporating the leakage power and the thermal-leakage loop in a DTM strategy forms the foundation of this article.

4 THE PROPOSED DTM STRATEGIES

4.1 Hybrid 3D + 2D Memory Architectures

Prior works [16, 39, 44, 46] have proposed hybrid 3D + 2D memory architectures (Figure 1). 3D memories (stacked 2D memory chips) offer several advantages in terms of area, bandwidth, and energy efficiency, which help to scale and meet the performance (bandwidth) requirements of modern-day processors. However, they are costly (higher cost/bit) and suffer from thermal issues arising out of higher power densities. Both these challenges can be addressed by integrating 2D memories with a 3D memory system (Figure 1), where 2D memories can serve the following two purposes:

(1) *Provide higher capacity:* The 3D memory is fast but has a higher cost per bit. A hybrid 3D + 2D memory system can provide adequate capacity at a reasonable cost.
(2) *Thermal management for 3D memory:* When 3D memory gets heated, data is migrated from 3D memory channels to 2D memory and accesses are served from 2D memory so that processors do not stall and continue making progress.

Therefore, many modern memory systems integrate both 2D memory (large but slow) and 3D memories (fast but small) to achieve both high capacity and bandwidth [16, 39, 44, 46]. Due to many-core systems and memory-intensive workloads (AI and ML applications) becoming popular, such hybrid memory architectures experience high utilization of bandwidth and capacity. Memory-intensive workloads keep 3D memory busy and idle periods are observed only during cooling. 2D memory can be used for DTM of 3D memory as well as to serve normal data accesses

from applications. Due to their large capacity, they experience a high utilization for workloads with a large memory footprint (common with AI/ML workloads). However, for compute-intensive workloads, the memories might experience a longer idle period or larger unused capacity, and state-of-the-art low power techniques could be used to reduce energy overheads.

In this work, we consider a hybrid 3D + 2D memory architecture and explore various approaches for managing 3D memory temperature by migrating data to 2D memory.

4.2 Thermal-Aware Migration

Temperature-dependent leakage power contributes significantly to 3D memory heating. To limit leakage power and meet thermal constraints, we propose using data migration at channel-level granularity and turning hot channels off. Traditional migration-based approaches that use a page-level granularity for power management suffer from several challenges: large overhead due to data migration delays, profiling and counting memory accesses per page, and prediction of appropriate low power state and power down timeout [33]. These challenges can be conveniently addressed for meeting thermal constraints of 3D memories due to the presence of multiple channels which enables faster data migration. The slow varying characteristic of temperature (compared to the clock period) allows sufficient time to implement a coarse-grained cooling strategy such as data migration. The coarse-grained channel-level profiling of accesses reduces the profiling overhead. The overhead of data migration delay is usually smaller than the high stall-time cost of cooling if data is not migrated, as shown in our experiments in Section 5.2.2.

Our proposed thermal-aware channel data migration (which runs every epoch of duration D) is defined in terms of temperature thresholds *T1*, *T2*, *T3*, and *T4* (*T4*>*T3*>*T2*>*T1*). We assume that the maximum temperature constraint *T4* (such as 80°C [31, 43, 44]) is specified for maintaining a safe temperature margin. Initially, each core is allocated a separate 3D memory channel to reduce memory interference [36]. Depending on the temperature, the thermal management scheme could be in any of the six states from *NORMAL* to *THROTTLE* (Figure 5). Below *T1*, the system works normally without any intervention. Above *T1*, data in the least accessed channels within the 3D memory is swapped with data in the central channels {5,6,9,10} (*SWAP* state). If the memory cools down below temperature *T1*, it transitions back to the *NORMAL* state; else, it moves to state *E1* (*E* refers to thermal emergency) where data is migrated from the interior 3D channels to 2D memory and these channels are turned off. The migrated data is accessed from the 2D memory until the 3D memory cools down to temperature *T1_C* which is lower than *T1* to prevent multiple transitions occurring around the threshold. The remaining 3D memory channels (which are not closed) are accessed normally without the processor being stalled. Also, since both 2D and 3D memories can operate in parallel, all cores are able to make progress.

On closing the interior channels, their temperature decreases, helping the adjacent channels to cool down. Now, the corner channels become the hottest as they are farthest from the interior channels. Hence, we close the diagonal corner channels {0,15} and {3,12} in sequence if the temperature rises to *T2* and *T3*, respectively. Above *T4*, accesses to channels {1,2,4,7,8,11,13,14} are data gated (accesses prevented) and channels {0,3,5,6,9,10,12,15} remain turned OFF/power gated until the 3D memory cools down to *T1* (*E1* state). We refer to the above algorithm as thermal-aware migration (TAM). TAM requires temperature sensors at the bottom layer which experiences the highest temperature and does not require temperature monitoring of the upper layers.

4.3 Memory Delay Model

Migrating data from (faster) 3D to (slower) 2D memory consumes 2D memory bandwidth and increases the queuing delay which could increase the time to complete memory requests even though both 2D and 3D memories operate in parallel. To avoid such adverse effects, we develop a delay

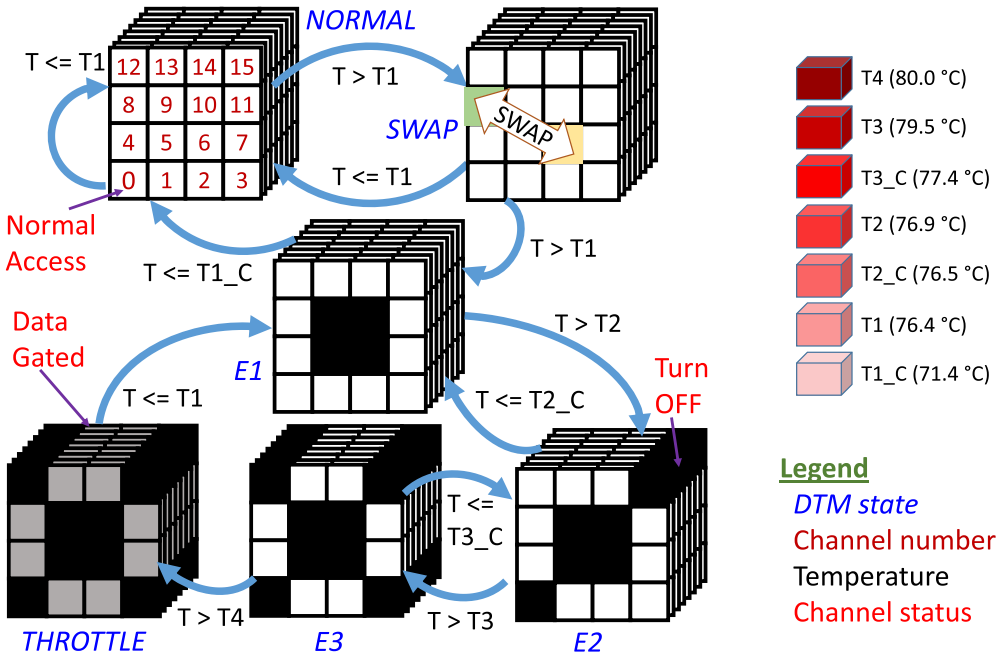


Fig. 5. Thermal-aware data migration and channel shutdown sequence.

model which estimates the time taken to complete 2D memory accesses, including the accesses for data migration. As the 3D memory heats up, we migrate data to (and subsequently access from) the 2D memory. The time to complete these requests consists of the data migration delay (*DMD*) and data access delay (*DAD*), which are impacted by both memory latency and bandwidth.

$$\text{Time to complete 2D memory requests} = DMD + DAD. \tag{1}$$

Data migration delay. Once the 3D memory is heated, we migrate s_{3D} (description in Table 1) bytes to 2D memory. When the memory is cooled down, we copy back s_{2D} bytes to the 3D memory. During migration, both memories operate in parallel (because of multiple channels), and we can write to 2D memory while reading the 3D memory. The 2D and 3D memories operate at bandwidth B_{2D} and B_{3D} with the slower of the two memories determining *DMD*.

$$DMD = (s_{3D} + s_{2D}) \times \max\left(\frac{1}{B_{3D}}, \frac{1}{B_{2D}}\right). \tag{2}$$

We obtain B_{2D} and B_{3D} using the following equations (memory is unavailable during refresh; hence available bandwidth reduces):

$$B_{3D} = b_{3D} \times n_{3D} \times (1 - \textit{3D Memory Refresh Overhead}),$$
$$B_{2D} = b_{2D} \times N_{2D} \times (1 - \textit{2D Memory Refresh Overhead}).$$

We calculate the *Refresh Overhead*, for the respective memory, using the following equation.

$$\textit{Refresh Overhead} = \frac{\textit{Time required for refresh}}{\textit{Refresh Interval}} = \frac{(\textit{Time required to refresh a row}) \times (\textit{Number of rows})}{(\textit{Refresh Interval})}.$$

Table 1. Parameter Definitions

Symbol	Description	Typical value
D	Time duration for an epoch	1 ms
C	Cache Line Size	64 Bytes
3D Memory Parameters		
b_{3D}	Per channel 3D memory bandwidth	8 GBps
L_{3D}	3D memory access latency	29 ns
s_{3D}	Size of data migrated, 3D → 2D	*Runtime*[1]
A_{3D}	Total accesses made to s_{3D} data (in last epoch)	*Runtime*[1]
n_{3D}	Number of 3D channels (migrating 3D → 2D)	*Runtime*[1]
2D DRAM Parameters		
b_{2D}	Per channel bandwidth	12.8 GBps
L_{2D}	2D memory access Latency	45 ns
s_{2D}	Size of data migrated, 2D → 3D	*Runtime*[1]
A_{2D}	Accesses made to 2D memory (in last epoch)	*Runtime*[1]
N_{2D}	Number of 2D memory channels	4
R_{2D}	Number of ranks per channel	2
BA_{2D}	Number of banks per rank	8
	Page Policy	Closed

[1] *Runtime* – Values are determined by DTM policy with the help of hardware counters.

Data access delay. We evaluate the impact of bandwidth (DAD_B) and latency (DAD_L) separately and add them to obtain $DAD = DAD_B + DAD_L$. Once data from hot channels is migrated from 3D to 2D memory, the total 2D memory accesses (A) increases by A_{3D} giving $A = A_{2D} + A_{3D}$. The total data fetched for A accesses and cache line size of C is $A \times C$. The delay due to bandwidth is given by $DAD_B = (A \times C)/B_{2D}$. DAD_L is the sum of the waiting time (when the memory is busy) in the queue (QD) and the memory latency delay (LD).

Latency delay. Each memory request issued to a memory bank requires L_{2D} time to be serviced, assuming a closed page policy. However, multiple requests can be active simultaneously at the multiple banks BA_{2D} in a rank. Similarly, multiple ranks R_{2D} and channels N_{2D} are active simultaneously, reducing the latency delay. We have

$$LD = \frac{A \times L_{2D}}{BA_{2D} \times R_{2D} \times N_{2D}}. \tag{3}$$

Queuing delay (QD). We incorporate formulations developed in Queuing Theory to model the waiting times for memory access requests at the queues. The memory (resource or server in queuing theory terminology) requests wait in a queue while the memory is busy. Queuing models are characterized by the input request (arrival process) distribution, service time (of the server) distribution, and the number of servers. Often, in queuing models, the arrival process and service time distributions are assumed to be Poisson and exponential, respectively, to simplify the analysis [14, 19, 47].

As our memory subsystem has a separate queue for each memory channel, we model each channel using an $M/M/1$ queue with multiple channels operating in parallel. An $M/M/1$ queue represents a system with a single queue per server, with the arrival (λ) and service rates (μ) assumed to be Poisson and exponential, respectively. The expected waiting time is given by $\frac{\lambda}{\mu \times (\mu - \lambda)}$. Assuming accesses are uniformly distributed across channels, the per channel access rate $\frac{(A \times C)}{(T \times N_{2D})}$

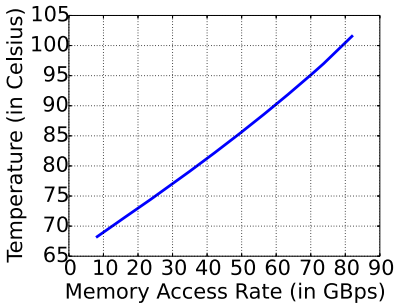


Fig. 6. Temperature exceeds 80°C with access rates >36 GBps.

and memory bandwidth (B_{2D}/N_{2D}) are the arrival (λ) and service rates (μ), respectively, giving

$$QD = \frac{A \times C}{T \times B_{2D}} \times \frac{N_{2D}}{B_{2D} - (A \times C/T)}. \tag{4}$$

The above analysis helps us estimate the time to complete the requests to 2D memory. If the total 2D memory delay is higher than a threshold, we can avoid migrating the data (Equation (7)). Using Equations (1)–(4), the total 2D memory delay is $\underbrace{DMD}_{\text{Migration Delay}} + \underbrace{DAD_B}_{\text{Bandwidth Delay}} + \underbrace{QD + LD}_{\text{Access Delay}}$.

4.4 FastCool

We now outline the FC algorithm, which aims to reduce unnecessary transitions in the TAM algorithm by adding certain checks to restrict the state transitions.

Applications with higher memory access rates lead to higher temperatures. We experimentally observe (parameters in Section 5.1) that the temperature rises above 80°C (Figure 6) for access rates higher than 36 GBps (~40K accesses per channel). Hence, as the temperature rises to *T1* (Figure 5), we transition to *E1* state only if the total access count of channels {5,6,9,10} exceeds A_{MIN} (for our architecture A_{MIN} is 160K, as four channels are closed and each channel's accesses should be higher than 40K).

$$\text{Minimum Access Rate condition: } A_{3D} > A_{MIN}. \tag{5}$$

This condition ensures that applications with predicted steady-state temperatures less than 80°C do not migrate data. To calculate A_{3D} in Equation (5), we require a count of accesses in the last epoch for each of the 3D memory channels, which we obtain using hardware counters (more details in Section 4.8.2).

As the 3D memory heats up, TAM migrates data without checking whether the 2D memory is overloaded. We must ensure that $\lambda < \mu$ to avoid overloading. From Equation (4):

$$\text{Queuing stability condition: } A < \frac{B_{2D} \times T}{C}. \tag{6}$$

TAM does not account for 2D memory speed before migrating hot data. Hence, a slow 2D memory could adversely affect the application execution time. To prevent such conditions, we set an upper bound on the estimated time to complete 2D requests. The upper bound on the delay (D_{MAX}) is found experimentally: we varied D_{MAX} and measured the performance for various workloads. We selected the D_{MAX} value that resulted in the minimum average execution time. For our architecture, this value is 8.415 ms (experimentally determined).

$$\text{Upper bound on 2D memory delay: } Delay < D_{MAX}. \tag{7}$$

Delay represents the total 2D memory delay caused due to data migration, bandwidth limitation, and access time, as discussed in Section 4.3.

Using Equations (5)–(7), FC ensures that the transition to higher thermal emergency states is made only for high access rates such that the migrated data does not saturate the 2D memory bandwidth and the estimated 2D memory delay is lower than a threshold value.

4.5 FC Policy Improvements

In FC, the channels to be closed for each DTM state transition are fixed at design time. We discuss a few insights obtained from deployment of FC as the DTM policy for 3D memory, that help the runtime selection of channels to be closed. These insights help us in identifying possible improvements in FC policy and subsequently lead to an improved DTM policy (EEFC).

4.5.1 Channel Temperature and Leakage Power. In FC, as the temperatures rise, the central channels are closed first, followed by the corner channels. Central channels get heated from all sides and exhibit the highest temperature if the accesses are spread uniformly across the channels. However, for non-uniform channel access, closing the central channels is not suitable as they might not exhibit the maximum temperature. Therefore, FC could be improved by incorporating dynamic selection of channels for closure and prioritizing channels with higher temperatures. Further, non-idealities in the manufacturing process can cause variability in channel leakage power and since the temperature is dependent upon leakage, we could also consider leakage power while choosing channels to be closed during DTM (more details on variability are presented in Section 6).

4.5.2 Channel Access Rate. As introduced in Section 1, FC migrates data to 2D DRAM and then closes the 3D memory channels to reduce temperature and leakage power. Subsequently, migrated data is accessed from 2D memory. However, migrating frequently accessed data from 3D to 2D memory increases the load on 2D memory, and thus, increases the 2D memory energy and the application execution time. Therefore, closing 3D memory channels with relatively lower access rates is a suitable choice to limit the overheads of DTM.

4.5.3 Adjacency. We use an additional insight that if a closed Channel A is horizontally or vertically adjacent to a hot Channel B, then Channel A provides a cooling surface to the neighbor B (even without closing Channel B). As an example, consider Channels 2, 3, and 7 as candidate channels for being turned off, with Channel 2 being at the highest temperature, and closed first (Figure 7(a)). To account for this adjacency condition, Channel 3 will not be closed since one of its adjacent channels (Channel 2) is already turned off. Instead, Channel 7 will be turned off as none of its adjacent channels is off. Such an approach helps realize a better spatial distribution of closed channels, and through that, of cooling surfaces.

4.5.4 Channel Position. Under uniform power dissipation, channels in the center are the most heated [34]. Thus, FC closes central channels to reduce heating. However, we can consider other closing orders as well. As shown in Figure 7(b), we prioritize closing of channels using a specific *order of closing*—(a) center, (b) corner, and (c) side channels, with the ordering of side channels being spatially distributed. Algorithm 1 describes the procedure to obtain the *OrderOfClosing* for a $P \times Q$ grid floorplan.

From these insights, we conclude that an efficient DTM strategy should consider temperature, leakage power (variability), access rate, adjacency, and position while closing the channels. This forms the basis of our three-pass DTM strategy, named EEFC, which is described next.

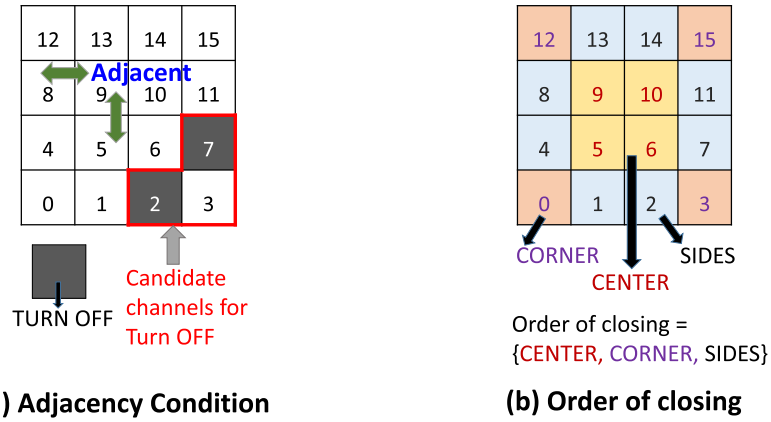


Fig. 7. Insights for improving the DTM policy.

ALGORITHM 1: Channel Closing Order

Input: P: Number of channels in rows (y-axis)
Input: Q: Number of channels in columns (x-axis)
Output: *OrderOfClosing*: A global list containing the order of closing

1 *CornerChannels*← $\{(0,0),(0,P-1),(Q-1,0),(P-1,Q-1)\}$
2 *BoundaryChannels*← $\{(x,y)|x=0$ OR $x=Q-1$ OR $y=0$ OR $y=P-1\}$
3 *SideChannels* ← *BoundaryChannels* − *CornerChannels*
4 *CentralChannels* ← {All channels} − *BoundaryChannels*
5 Re-arrange each of *CornerChannels*, *SideChannels*, *CentralChannels* such that adjacent channels in the respective lists are not at consecutive locations in the floorplan.
6 *OrderOfClosing* ← [*CentralChannels*, *CornerChannels*, *SideChannels*]

4.6 Energy-Efficient FastCool

The EEFC strategy is built upon FC and uses the data migration approach to manage temperature and a state sequencing (Figure 5) similar to FC. However, unlike FC which decides the channels to be closed at design time itself, EEFC adds more sophistication to FC and decides the channels to be closed at runtime. Similar to FC, as the memory heats up, EEFC transitions to higher DTM states and closes additional memory channels. We show the DTM states of EEFC in Table 2. There are five DTM states named *E0–E4*. *E0* is the initial state without any channels being closed and corresponds to the *NORMAL* state in Figure 5. *E4* corresponds to the *THROTTLE* state. EEFC does not use the *SWAP* state in the DTM state machine (Figure 5). A pre-defined number of channels (Nx) is closed in each of these states (Ex). While heating, each state (Ex) has a temperature threshold (Tx) at which the system enters Ex. The system exits state Ex when the temperature falls below the cooling threshold, Tx_C ($<Tx$).

Similar to prior works [31, 43], we assume a maximum temperature constraint of 80°C (= $T4$). We use $\{N1, N2, N3, N4\} = \{4, 6, 8, 8\}$ similar to FC. Whenever the system moves to higher DTM states, a fixed number (K) of channels are closed. When the 3D memory cools down, channels with the lowest temperature are opened, and the corresponding data is migrated back from 2D to 3D memory. Channels at the lowest temperature are opened as they can be accessed longer before getting heated. The channels to be closed (total count = K) are selected based on the proposed three-pass approach described as follows.

Table 2. Different DTM States in EEFC

State	Number of channels closed	Threshold during heating (°C)	Threshold during cooling (°C)	Memory stall?
E0	0	*T0*	-	No
E1	*N*1	*T1*	*T1_C*	No
E2	*N*2	*T2*	*T2_C*	No
E3	*N*3	*T3*	*T3_C*	No
E4	*N*4	*T4*	*T4_C*	Yes

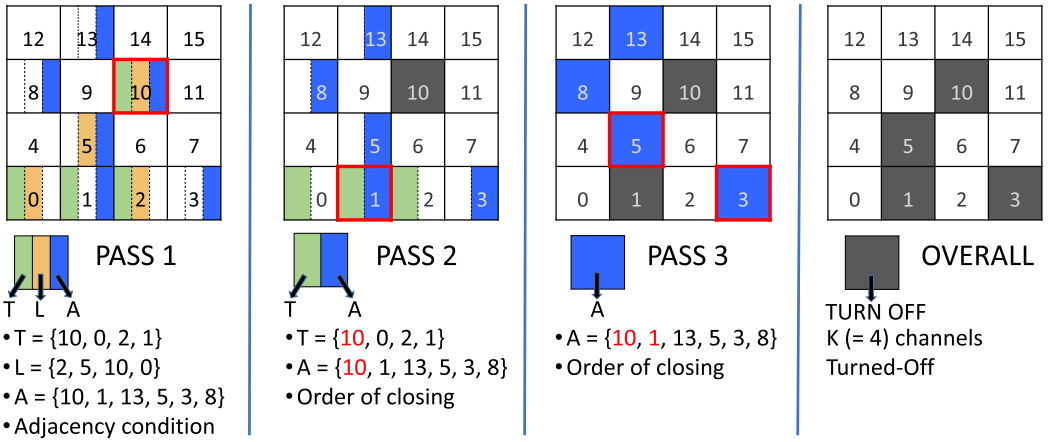


Fig. 8. *EEFC DTM policy*: Three-pass approach for selecting the channels for closing: T, L, A represent the channels with high temperature, high leakage, and low access rate, respectively. A box (channel) is considered for closing only if it is completely filled using colors indicated in the figure.

4.6.1 Pass 1. The first pass attempts to close the channels with high temperature, T, and high leakage power, L (Section 4.5.1). In this regard, a channel is selected for closing only if it is one of the top K channels in terms of both temperature and leakage power. While in an ideal case, channels with higher temperature would have higher leakage power, the relationship may not hold in the presence of process variations. Therefore, we consider both temperature and leakage power to make EEFC resilient to process variations (details in Section 6.3.1). Also, the selected channel is closed only if it has a lower access rate relative to the overall access rate of the 3D memory (Section 4.5.2). We classify the access rate for a channel to be low if it is less than $ACC\_MULT \times AAR$, where AAR is the average access rate across all the channels and ACC_MULT is an experimentally determined constant. Moreover, we do not close adjacent channels as an adjacent closed channel would already act as a cooling surface (Section 4.5.3).

The selection of channels to be closed by EEFC is shown in Algorithm 2. The algorithm identifies a set of low access rate channels in Lines 1–5. In Pass 1 (Lines 7–14), we select channels with high temperature, high leakage, and low access rate (Line 7) as candidates for closure and check for adjacency in Line 9 before closing them.

In Figure 8, T and L represent the top K (=4) channels in terms of temperature and leakage power respectively, while A represents the channels with low access rate. In Pass 1, we mark the channels with appropriate colors if they belong to T, L, or A. A channel is closed if its square box

ALGORITHM 2: Channel shutdown strategy

Input: N_C: Total number of channels in 3D memory
Input: K: Number of channels to be closed
Input: T: {Top K channels in terms of temperature}
Input: L: {Top K channels in terms of leakage power}
Input: ACC_MULT: Experimentally determined constant multiplier for access rate
Input: $A_C[0..N_C\text{-}1]$: Access rate for all channels
Output: C_{off}: Set of K channels turned off

```
   /* Identify channels with low access rate                          */
1  AAR ← average(A_C)
2  A ← φ                                  /* Set of channels with low access rate */
3  for i ← 0 to N_C − 1 do
4      if A_C[i] < ACC_MULT × AAR then
5          A ← A ⋃ {i}
       end
   end
   /* Pass 1 starts                                                   */
6  k ← 0
7  C ← T ∩ L ∩ A                                  /* Candidates for closure */
8  foreach c ∈ C do
9      if no channel adjacent to c is OFF then                  /* Adjacency check */
10         closeChannel(c)                                      /* Close channel */
11         C_off ← C_off ⋃ c
12         k ← k + 1
       end
13     if k = K then
14         return C_off
       end
   end
   /* Pass 2 starts                                                   */
15 C ← T ∩ A                                  /* High temperature,low access rate */
16 for i ← 0 to N_C − 1 do
17     c ← OrderOfClosing[i]
18     if c ∈ C and c is not closed then
19         closeChannel(c)
20         C_off ← C_off ⋃ c
21         k ← k + 1
       end
22     if k = K then
23         return C_off
       end
   end
   /* Pass 3 starts                                                   */
24 C ← A                                      /* Low access Rate channels */
25 ... Remaining portion for Pass 3 is same as lines 16–23 in Pass 2 above
```

is fully colored (i.e., present in all of T, L, and A), and it is not adjacent to a closed channel. In Figure 8, only Channel 10 is wholly filled with colors and is chosen for closing in Pass 1.

4.6.2 Pass 2. During Pass 1, we might not be able to close the required number of channels (K) due to the multiple conditions not being satisfied. For instance, in Figure 8, we closed only one channel in Pass 1 compared to the desired 4. Therefore, we relax some conditions in Pass 2. We consider channels for closing in a pre-defined order based on their position (Section 4.5.4). Additionally, the channel chosen for closing must have a low access rate (A) and must be one of the top K channels when sorted in decreasing order of temperature (T). Lines 15–23 of Algorithm 2 describe Pass 2. We select high-temperature, low access rate channels for closure and perform an *OrderOfClosing* check. In Figure 8, for Pass 2, Channels 0, 1, and 2 have high temperatures, out of which only Channel 1 has low access rate (Channel 10 is already closed in Pass 1). Therefore, Channel 1 is selected in Pass 2 for closing.

4.6.3 Pass 3. At the end of the second pass, we still might not be able to close K channels as high-temperature channels might not exhibit lower access rates. Thus, we relax the temperature condition in the third pass and just consider the order of closing and the access rate of channels to select the remaining number of channels for closing (Algorithm 2, Line 24). As shown in Figure 8, we follow the order of closing and select Channels 5 and 3 for closing (both with low access rates), thereby closing a total of K (=4) channels (1, 3, 5, 10) in the three passes.

While EEFC overcomes some of the limitations of FC, it is a relatively complex policy. EEFC determines channels to be closed at runtime using three passes. It also requires estimation of leakage power for each channel at runtime. This additional complexity of EEFC provides a significant reduction in memory energy consumption and energy-delay product (EDP), which we present as results in Section 5.

4.7 Leakage Current Estimation

EEFC needs to compute the leakage power per channel (temperature dependent) in Pass 1, which we calculate as follows. The channel temperature at the base layer is obtained through temperature sensors and is monotonically decreasing from the base layer toward the top [11]. From our experiments (discussed in Section 5.3.2), we observe that the temperature across the vertically aligned ranks within a channel follows an approximately linear pattern and use this observation to estimate the temperature of inner ranks within each channel (Figure 20) as a linear interpolation of the base and top layer (heat sink) temperatures. The estimated temperature is used to calculate the leakage power of each rank. The channel-level leakage power is the sum of the rank-level leakage power values over all the ranks in a channel.

In Section 6, we extend the above temperature-dependent leakage current estimation procedure to account for process variations and each rank's leakage power is scaled, corresponding to the process variation experienced by the rank during manufacturing (details in Section 6.2).

4.8 DTM Policy Implementation

The DTM policy is implemented as a part of the runtime management software (operating system) and executes on a timer interrupt every 1 ms. The software monitors the 3D memory temperature and channel-wise access counts for FC and EEFC, and if it is heated beyond a threshold, it selects the channels for closing. Before closing, the data of selected 3D memory channels is migrated to 2D memory using DMA. The migration for different channels can occur in parallel as memory channels are independently accessible. Since each core uses a separate 3D memory channel to reduce memory interference (similar to previous works [31, 36]), only the cores for which data is being migrated are stalled. The rest of the cores continue to make progress. To simplify the migration and

access of data from 2D memory, we consider a 3D memory channel mapped to a specific 2D memory channel and 2D memory channels to have a fixed space allocated for accommodating the migrated data (if there is no migrated data, the fixed space can be used to serve normal data accesses).

A remap table is maintained in the memory controller (3D memory) to allow access to migrated data. For each 3D memory channel, the remap table contains an entry to indicate if the channel is migrated or not and the corresponding 2D memory channel number where the data is migrated. Subsequent accesses to migrated data are served from the 2D memory with the help of the remap table. When the 3D memory cools down, data is brought back from the 2D memory using steps similar to the above. Also, the remap table data migration flag is updated, and accesses are serviced from the 3D memory.

We discuss the remap table overheads and access counter overheads in the following text.

4.8.1 Overheads of Remap Table. The DTM policy requires a small remap table (hardware), maintained in the memory controller of the 3D memory, to support redirecting the accesses for migrated channels. For each 3D memory channel, the remap table has two entries: (i) *migrated flag* (1 bit): to indicate if the 3D memory channel is migrated (Set) or not (Reset) and (ii) *2D memory channel number*: to locate data in 2D memory (if migrated). In our architecture, we use a 16-channel 3D memory and a 4-channel 2D memory (2 bits). So, each entry is 3 bits (1 bit for the migrated flag and 2 bits for the 2D channel number), and the overall table size is 48 bits (= 16 channels $\times$ 3 bits).

Note: We assume that 2D memory has fixed addresses/space allocated for accommodating the migrated data. When there is no migrated data from the 3D memory (the migrated flag is Reset), the addresses/space can be used to serve normal data accesses from the 2D memory.

4.8.2 Overheads of Channel-Level Access Counters. The EEFC and FC policy require a count of accesses in the last epoch at channel-level granularity for both 3D and 2D memories to calculate A_{3D}, A_{2D}, respectively, which are used for estimation of 2D memory delay/load. We use hardware counters to keep track of the access counts.

To determine the access counter's size, we need to know the maximum number of accesses that can occur in an epoch, which is ascertained using the maximum channel bandwidth. For example, in our architecture, the channel bandwidth for 3D memory is 8 GBps, and epoch time is 1 ms. Thus, a maximum of 8 MB data can be accessed in 1 ms. To fetch 8 MB data, we need 128K access (each access is 64 bytes). So, we need a 17-bit counter (2^{17} = 128K). Since we consider 16 memory (3D) channels in this work, we need 272 bits for the access counters. Similarly, for a four-channel 2D memory with a per-channel bandwidth of 12.8 GBps, we require an 18-bit access counter and 72 bits (=18 bits $\times$ 4 channels) in total.

Thus, the DTM policy requires a 48-bit remap table and 344 bits (=272 + 72) for access counters. Moreover, the DTM policy has a software runtime overhead of less than 2 μs on the simulated architecture (Table 3; more architecture details are presented in the next section).

5 EXPERIMENTS AND RESULTS

5.1 Experimental Setup

For evaluating the proposed algorithm, we use a trace-based framework. We run the workload on the Sniper [5] performance simulator and collect a memory access trace assuming only 3D memory is present. The core and memory architectural parameters are shown in Tables 1 and 3 (similar to [34, 43]). Using the energy-per-access values (20.55 nJ per 64 bytes access [43]) determined from CACTI-3DD [8], we convert the access trace to a power trace (with 1 ms epochs) which is fed to the HotSpot 6.0 [51] thermal simulator. We add the refresh power and the temperature-dependent leakage (Figure 3) to the power trace to obtain the total power [44]. These modifications to HotSpot

Table 3. Core Architecture Parameters

Parameter	Value
Number of Cores	16
Core Model	2.1 GHz, 2.1 V, 45 nm, out-of-order, 3-way decode, 84 entry ROB, 32 entry LSQ
L1 I/D Cache	64 KB@2 ns, 2-way/64B-block
L2 Cache	Private, 512 KB@6 ns, 16-way/64B-block
3D Memory Configuration	1 GB, 16 channels, 8 ranks, 1 bank per rank, closed page policy
3D Memory Timing	29 ns (latency), 8 GBps (per channel bandwidth)
2D Memory	Details in Table 1

Table 4. HotSpot Configuration Values

Parameter	Value
Chip thickness	0.1 mm
Silicon thermal conductivity	100 W/mK
Silicon specific heat	1,750 kJ/m^{3}K
Spreader thickness	1 mm
Spreader thermal conductivity	400 W/mK
DRAM thickness	0.02 mm
DRAM thermal conductivity	100 W/mK
Heatsink thickness	6.9 mm
Heatsink resistance	0.1 K/W

Table 5. Workloads and Benchmark Details

Benchmark Details and Type	Workload Name
povray(×16) – Compute	*povray*
perlbench(×16) – Compute	*perlbench*
bwaves(×16) – Mixed	*bwaves*
GemsFDTD(×16) – Mixed	*GemsFDTD*
soplex(×16) – Memory	*soplex*
lbm(×16) – Memory	*lbm*
povray(×4), perlbench(x4), soplex(×4), lbm(x4)	*ppsl*
povray(×4), perlbench(×4), bwaves(×4), GemsFDTD(×4)	*ppbg*
bwaves(×4), GemsFDTD(×4), soplex(×4), lbm(×4)	*bgsl*
perlbench(×4), lbm(×8), GemsFDTD(×4)	*pllg*

are validated against ANSYS Icepak [1] and the transient temperature has been found to be accurate within 1°C [44]. A high-level modeling for the 2D memory including queuing delay and data migration is implemented so as to account for the overheads. Temperature thresholds for TAM and FC are $\{T1, T2, T3, T4\} = \{76.4, 76.9, 79.5, 80\}$ and $\{T1\_C, T2\_C, T3\_C\} = \{71.4, 76.5, 77.4\}$ (in Celsius, determined experimentally). Further, for EEFC, we obtain ACC_MULT=1.1, $\{T1, T2, T3, T4\} = \{76.4, 77.1, 79.5, 80\}$ (in°C), and $\{T1\_C, T2\_C, T3\_C\} = \{71.4, 76.5, 77.6\}$ (in°C). We use our simulation-based setup to determine various parameters and thresholds experimentally (discussed in Section 5.3). We vary the parameters of interest, record the EDP, and select the parameter values with minimum average EDP across workloads.

Configuration parameters for HotSpot are shown in Table 4 (similar to [31, 44]). For cooling, we use the default air-cooled heat sink model as provided by HotSpot 6.0 and also used by the Baseline policy [31]. We use benchmarks from SPEC CPU2006 suite for evaluation.[1] We classify the SPEC CPU2006 benchmarks into three categories and present experimental results for two representative benchmarks from each category: compute-intensive (*povray*, *perlbench*), mixed (*bwaves*, *GemsFDTD*), and memory-intensive (*soplex*, *lbm*). Sixteen instances of each

[1]Limaye and Adegbija [29] observed the memory and cache access rates to be similar for benchmarks in both SPEC CPU2006 and SPEC CPU2017 suites. Hence, we expect similar thermal issues and identical benefits for 3D memories when using either of the benchmark suites.

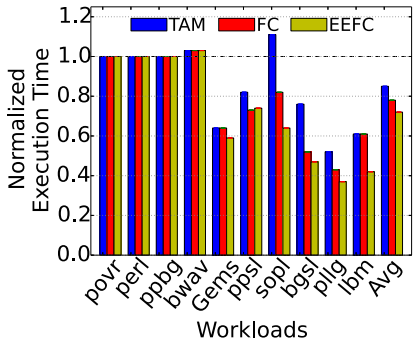


Fig. 9. Normalized execution time comparison.

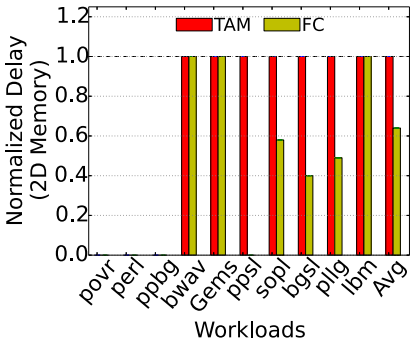


Fig. 10. 2D memory delay comparison.

representative benchmark, one instance on one core, form the first six workloads (Table 5). Mixes of the representative benchmarks are used to get the next four workloads, building a total of 10 workloads. The workloads are simulated until each benchmark has completed at least 1 billion instructions, restarting the benchmark(s) that complete early.

5.2 Results and Discussion

5.2.1 Execution Time Comparison. Figure 9 shows the execution time of the proposed strategies TAM, FC, and EEFC normalized with respect to the baseline. We use the 3D memory page allocation [31] as a baseline for comparison as it is the closest related work; here, the authors assume only 3D memory is present and redirect page allocation requests for the hot channel to the coldest channel. If the coldest channel is full, space is created within it by moving its pages to other cold channels. The baseline strategy operates on workloads with a temperature higher than 75°C [31]. However, the reallocation reduces neither dynamic nor leakage power. Hence, the memory has to stall and cool down frequently. Further, page swaps also increase the execution time. At high temperatures, leakage values are comparable to dynamic power. Thus, it is hard to ensure a fast cooldown with throttling/data gating alone.

TAM, FC, and EEFC migrate data in hot channels from 3D to 2D memory and then turn the channels off; this reduces leakage power and also reduces the number of times the system is throttled. Since both 2D and 3D memory operate in parallel, the migrated data can be accessed from the 2D memory ensuring all the applications make progress. Also, EEFC closes fewer channels than FC due to its runtime decision making, thereby reducing the data migrations to/from 2D memory, resulting in a better performance. Overall, due to channels being turned OFF in the proposed approach, the 3D memory gets cooled down much faster than the Baseline approach, which needs to stall longer and frequently to achieve cooling. Therefore, even though there are migration overheads, we observe execution time reductions (FC versus baseline) of 22% on average and a maximum of 57%. EEFC provides an additional performance improvement of 8% on average and up to 30%.

FC and TAM can achieve comparable execution time. However, FC places a lower burden on the 2D memory than TAM. As shown in Figure 10, using the FC strategy, the time to complete 2D memory requests reduces by 36% as compared to TAM. The minimum access rate condition (Equation (5)) reduces unnecessary transitions and migrations to 2D memory.

5.2.2 Temperature and State Transition Comparison. We visualize the operation of various DTM approaches in more detail using a temperature-time trace and a state transition plot for the *bgsl*

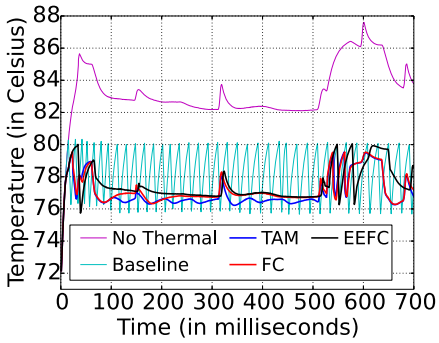


Fig. 11. Temperature-time trace for *bgsl* workload.

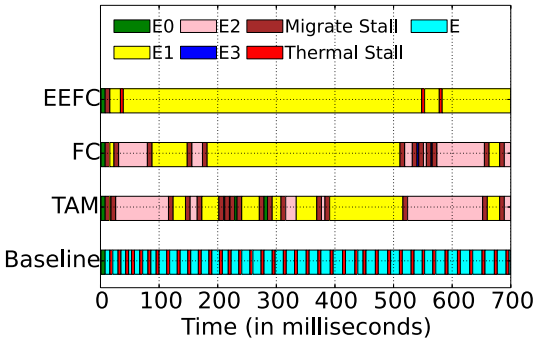


Fig. 12. DTM state transitions for *bgsl* workload.

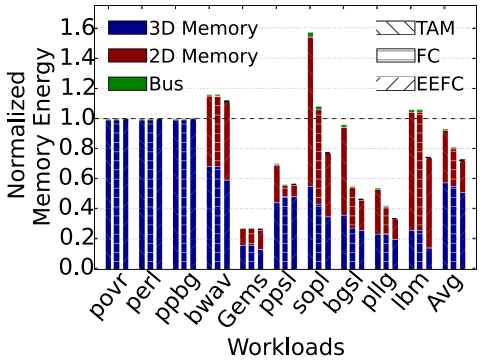


Fig. 13. Normalized memory energy (3D + 2D) comparison.

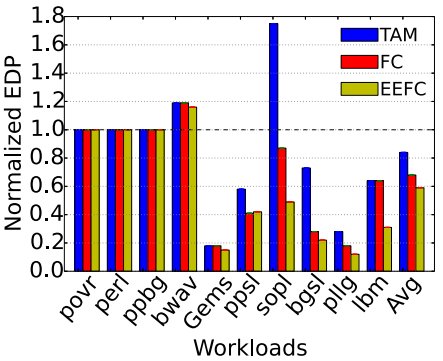


Fig. 14. Normalized EDP comparison.

workload. As shown in Figure 11, if there were no thermal constraint, the temperature would exceed 80°C. The baseline strategy throttles at 80°C until it cools down to 76.4°C in a few milliseconds (time constant similar to prior works [2, 10, 27, 28, 49]). However, TAM and FC can reduce leakage and dynamic power; hence, they eliminate thermal stalls and are able to maintain the temperature in the range of 76–79.4°C.

From the temperature-time trace and state transition plot (Figure 12), we observe that TAM closes channels aggressively and maintains a temperature of about 76.9°C, frequently migrating between *E0*, *E1*, and *E2* states. However, FC does not close channels at T2 (76.9°C) because of the minimum access rate condition (Equation (5)); hence, its temperature rises to 79.4°C. At around 500 ms, due to higher access rates in the corner channels, FC transits to *E2* and then to *E3*. Compared to TAM, FC effectively utilizes the temperature headroom and does not cause unnecessary transfers to the 2D memory. EEFC remains in *E1* for almost the entire execution of the *bgsl* workload. It further reduces the number of state transitions by appropriately selecting the channels to be closed at the runtime. EEFC is able to operate at even higher temperatures than FC, closer to 80°C, and efficiently utilizes both 3D and 2D memories.

5.2.3 Energy Comparison. EEFC's flexibility in the selection (and order) of closing channels results in lesser data movement between 3D and 2D memories, and keeps the 2D memory turned off for longer times, leading to lower energy overheads with respect to TAM, FC, and the baseline (Figures 13 and 14). Using FC, we observe an average improvement of 19% and 32% (up to 72%

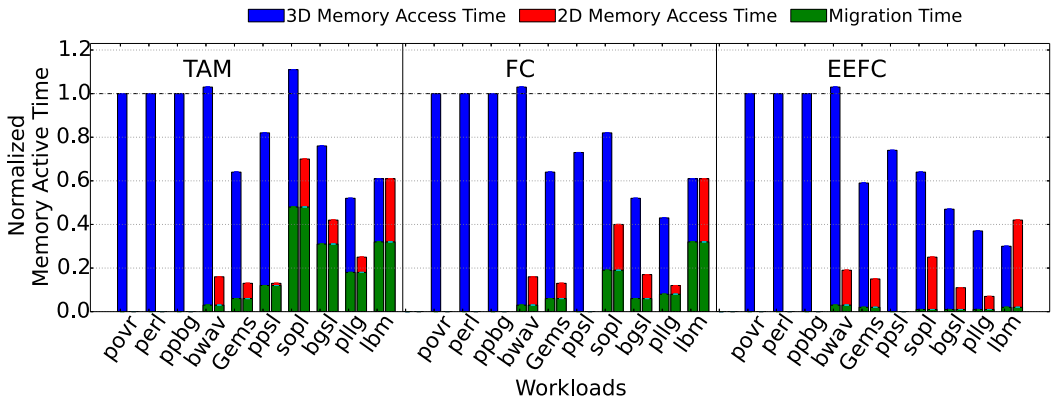


Fig. 15. Active time comparison for various policies normalized with respect to Baseline policy.

and 82%) in memory system energy consumption and EDP, respectively, over the baseline. EEFC provides an additional benefit of 10% and 16% (up to 30% and 51%) in memory system energy consumption and EDP, respectively, over FC.

For the *bwaves* workload, the transient temperature does not reach 80°C. However, the reactive thermal management policies start acting earlier so that temperature does not overshoot the operating limit. As temperature rises, TAM, FC, and EEFC migrate data to the 2D memory consuming significant energy as compared to the baseline policy, which does only page allocation for 3D memory. Hence, for *bwaves*, we see higher memory energy consumption and EDP for the proposed DTM policies.

For the *soplex* workload (abbreviated to *sopl* in the figures), the TAM and FC policies close additional 3D memory channels as compared to EEFC, increasing 2D memory load and memory energy consumption. However, EEFC is able to select appropriate channels for closure by considering temperature, access rate, and adjacency, and reduces memory energy consumption by 33%, whereas TAM increases energy consumption by 58% as compared to the baseline.

5.2.4 Data Migration Overheads. The performance and energy consumption of the proposed DTM policies is affected by data migration. To quantify the migration overheads, we study and compare the active time and memory energy consumption of 2D/3D memories for various workloads.

Comparison of active times of 3D and 2D memory for various workloads. In Figure 15, we plot the time for which the 3D and 2D memories are active (includes migration and access times), normalized to Baseline policy. For 2D memory, we show active time only for migrated data from the 3D memory. For the compute-intensive workloads (*povr*, *perl*, *ppbg*), 3D memory does not get heated beyond 76°C. Thus, there are no migrations to the 2D memory. For *bwav*, as temperature rises above 76.4°C, the proposed strategy migrates data to 2D memory. Similarly, in other mixed workloads (*Gems*, *ppsl*), we see a small number of accesses serviced by the 2D memory. Memory-intensive workloads (*sopl*, *bgsl*, *pllg*, *lbm*) lead to higher temperatures and data transfer to (and subsequently, access from) the 2D memory.

TAM frequently migrates between various thermal emergency states resulting in larger 2D memory load and time spent in migration. However, FC and EEFC avoid unnecessary migrations reducing both migration and access time. Also, EEFC selects channels to be closed dynamically, reducing the state transitions significantly, and achieves the lowest active and migration time. On an average, TAM, FC, and EEFC spend 20%, 12%, and 2% of the active time in migrations.

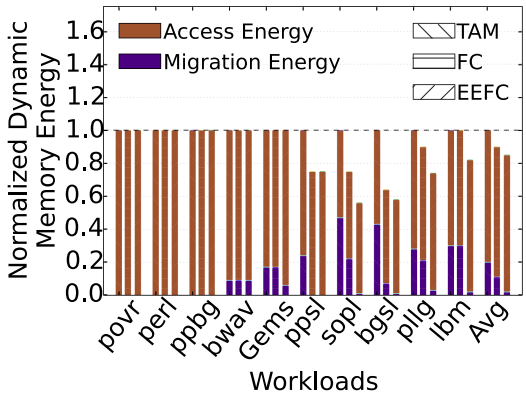


Fig. 16. Normalized dynamic memory energy comparison.

Comparison of dynamic memory energy for various workloads. As shown in Figure 16, both migration and access contribute to dynamic energy. On average, FC decreases dynamic memory (2D + 3D) energy by 10% as compared to TAM. EEFC provides an additional average dynamic memory (2D + 3D) energy reduction of 6%. Reduction of unnecessary state transitions results in saving of dynamic memory and migration energy. The migration energy contributes to 20%, 12%, and 3% of the dynamic memory energy, and 6%, 4%, and 1% of the total memory energy for TAM, FC, and EEFC, respectively.

5.3 Determining DTM Parameters

In this section, we discuss our methodology used for determining various tunable parameters for the proposed DTM policies.

5.3.1 Temperature Thresholds. We divide the set of workloads into two parts: training and test, both containing compute-intensive, memory-intensive, mixed benchmarks, and mixes of the above benchmarks. Specifically, the training and test workloads are {*perlbench*, *bwaves*, *ppsl*, *pllg*, *lbm*} and {*povray*, *GemsFDTD*, *soplex*, *ppbg*, *bgsl*}, respectively. We vary the parameters of interest and parameter values that result in minimum average EDP across training workloads are selected. We verify the effectiveness of the selected values on the test workloads.

We illustrate the above procedure through the determination of *T3* for the TAM algorithm. We show EDP results for $T3 = \{78, 79, 79.5, 79.7\}$ (in°C) on the training and test workloads in Figures 17 and 18, respectively. The EDP results are normalized with respect to $T3 = 79.5$°C. As seen from Figure 5, when the temperature crosses *T3*, TAM transitions from DTM state *E2* (with six closed channels) to *E3* (with eight closed channels). Further, a lower value of *T3* causes an early transition to *E3*, closing additional channels and migrating data to 2D memory, thereby increasing the memory energy consumption. Therefore, for $T3 = 78$°C, we observe a rise in EDP for memory-intensive workloads (Figure 17). Compute-intensive workloads and some mixed workloads are unaffected as they do not reach temperatures greater than 78°C. Also, for $T3 = 79$°C, we see a rise in EDP for the *pllg* workload (in the training workload set).

If the *T3* value is high and close to 80°C, the memory does not spend enough time in *E3* state and quickly transitions to *THROTTLE* state. This is because the DTM action of closing two additional channels takes time and also, cooling is a slow phenomenon. Therefore, temperature continues to rise and crosses 80°C. In such cases, the DTM policy is unable to utilize the *E3* state effectively and overcompensates by remaining in the *THROTTLE* state for longer times. Therefore,

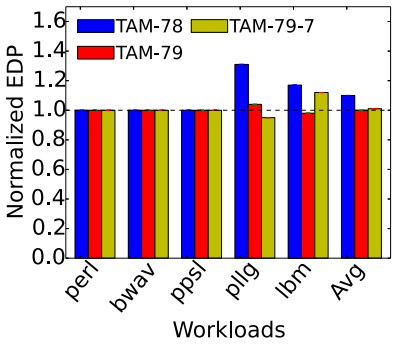


Fig. 17. Normalized memory EDP comparison for training workloads.

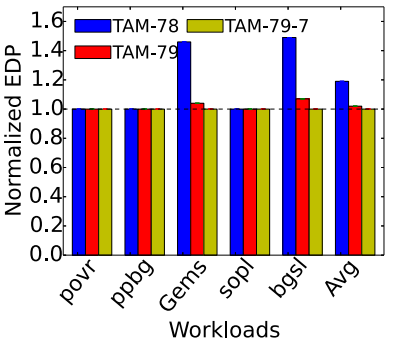


Fig. 18. Normalized memory EDP comparison for test workloads.

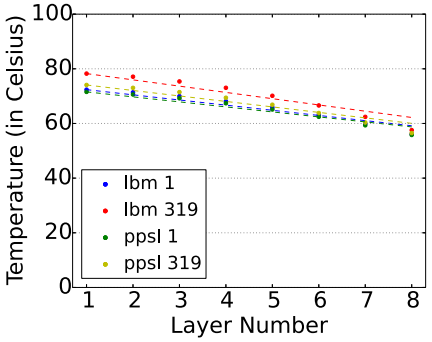


Fig. 19. Layer-wise temperature (Layer 8 is top layer and closer to heat sink).

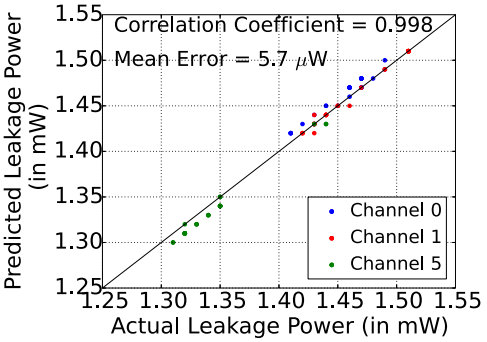


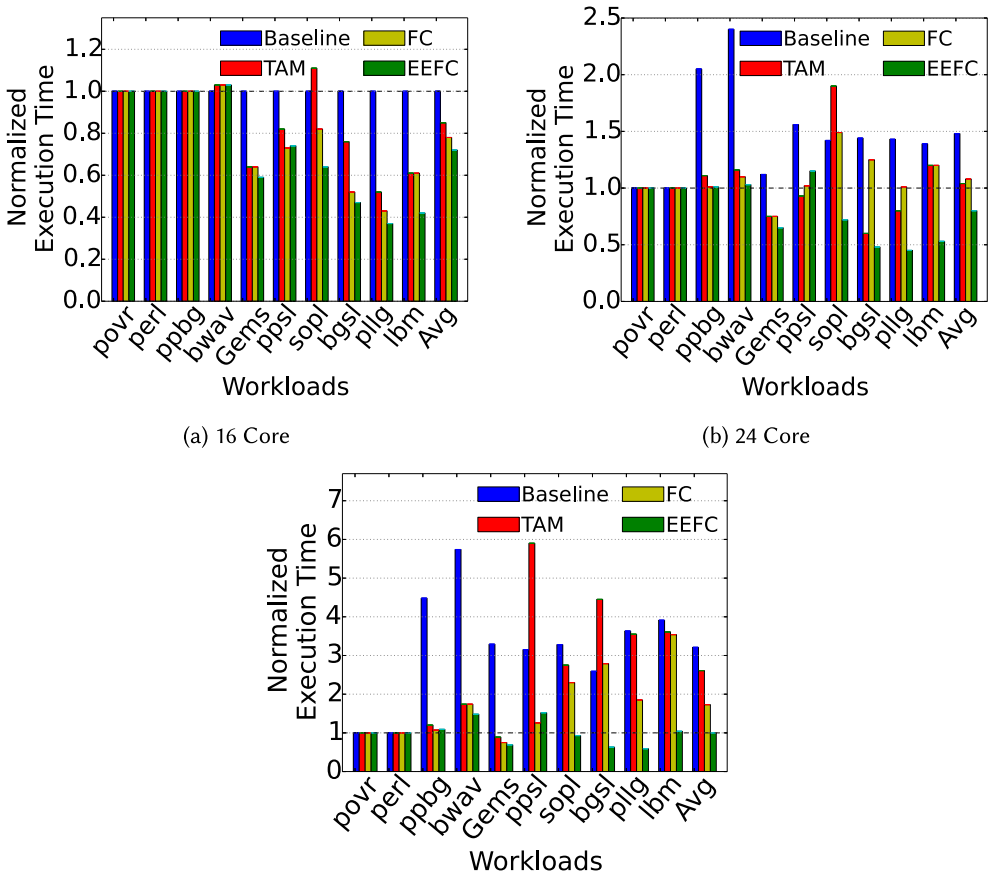
Fig. 20. Actual versus estimated leakage using linear temperature prediction.

for $T3 = 79.7°C$, we see a rise in EDP for memory-intensive workloads (such as *lbm*). We choose $T3 = 79.5°C$ as it results in minimum average EDP for the training workload set. We observe similar behavior in the test workloads as well.

We follow a similar approach to determine the value of other temperature parameters and access rate thresholds.

5.3.2 Leakage Power Determination for EEFC. Figure 19 shows the temperature for each layer for Channel 1 when executing training workloads ({*ppsl*, *lbm*}) at times 1 ms and 319 ms (the temperature trend is similar for other timesteps and channels as well). We observe that the rank temperature follows an approximately linear pattern within a channel. Hence, we fit a linear model to predict the temperature of internal ranks, given the bottom layer rank temperature and the heat sink temperature. Using this linear model, we observe a higher error in the estimated temperature at lower temperatures. However, since the effect of temperature on leakage power is less at lower temperatures, the model is still able to provide a good estimate for the leakage power.

We use the predicted temperature of a rank to calculate the rank's leakage power (the leakage power trend with respect to temperature was obtained using CACTI 3DD; see Figure 3). Figure 20 shows that the estimated leakage power based on linear temperature pattern is very accurate in comparison to the actual leakage power. In Figures 19 and 20, the actual temperature and leakage power values are obtained from the HotSpot thermal simulator. As mentioned earlier, we have



(a) 16 Core

(b) 24 Core

(c) 32 Core

Fig. 21. Execution time comparison for various policies normalized with respect to Baseline policy for 16-core system.

updated HotSpot (Section 5.1) to model the effect of temperature-dependent leakage power, and thus, we could monitor leakage power values during simulation.

For Channels 0 (corner), 1 (side), and 5 (center), we obtain a correlation coefficient of 0.998 in the rank-level leakage power estimation for different workloads ({*soplex*, *bgsl*, *pllg*}) at various timesteps. The correlations are similar for other workloads and channels as well. Such a leakage power estimator allows us to accurately determine the temperature-dependent leakage of channels at runtime.

5.4 Sensitivity Analysis

We discuss the sensitivity analysis of the various DTM policies, studying the impact of certain key parameters on execution time.

5.4.1 Varying Number of Cores. We evaluate performance for the Baseline [31], TAM, FC, and EEFC DTM policies for various systems sizes (number of cores): 16 (default), 24 (1.5×), and 32 cores (2×).

For 16 cores (Figure 21(a)), we observed execution time reductions (FC versus Baseline) of 22% on average and a maximum of 57%. EEFC leads to an additional performance improvement of 8% on average and up to 30% compared to FC. For *bwav*, we observe that our proposed strategies (TAM, FC, and EEFC) have higher execution time than the Baseline approach for the 16-core configuration. The proposed policies trigger DTM operation early at 76.4°C and migrate data to 2D memory. However, eventually, *bwav* temperature does not rise to 80°C. This unnecessary data migration stall increases the execution time. In the case of *sopl*, we observe TAM has higher execution time than the Baseline policy as it frequently transits between *E0* and *E1* state, significantly increasing migration stalls and overall execution time. FC and EEFC do not migrate for low access rates and avoid unnecessary transitions/stalls, thus achieving better execution time than Baseline and TAM for *sopl*.

For 24 cores, we increase the number of instances of benchmarks. As an example, for the *lbm* workload, we run 24 instances of *lbm* benchmark; for *ppbg* workload, 6 instances each of povray, perlbench, bwaves, and GemsFDTD are executed, and we modify the other workloads similarly. This increase in the memory load increases the execution time for all DTM policies. We observed execution time reductions (FC versus baseline with 24 cores) of 22% on average and a maximum of 54%. EEFC leads to an additional average performance improvement of 33% and up to 61% compared to FC.

The temperatures for *bwav* and *ppbg* start exceeding 80°C. The Baseline policy is unable to reduce power and stalls frequently, which reduces performance significantly (especially for *bwav* and *ppbg*). In the case of *sopl*, TAM and FC policies have frequent state transitions between *E2* and *E3*, which cause data migration stalls and reduce performance. The FC policy decides channels to be closed statically. In many cases, for 24 cores, it estimates a high 2D memory load and avoids migrations. It frequently stalls for cooling, reducing performance. EEFC decides channels to be closed dynamically and is appropriately able to determine on stalling or migration (except for *ppsl*) and scales well, achieving the lowest execution time.

As we increase the core count to 32 (see Figure 21(c)), EEFC scales well. However, the rest of the policies suffer large overheads. We observe that EEFC is able to reduce execution time by 78% compared to the Baseline approach.

5.4.2 Varying Number of 2D Memory Channels. We study the effect of changing 2D/3D memory units by varying the number of 2D/3D memory channels. When the 2D and 3D memory units increase in the same proportion, the performance results are similar. When we increase the number of 2D memory channels ($N_{2D} = \{2, 3, 4\}$) keeping the 3D memory units fixed, we observe that there is no change in execution time for compute-intensive and mixed workloads as these benchmarks migrate less amount of data. Memory-intensive workloads such as *lbm* show a considerable reduction in execution time for a larger number of 2D memory units as shown in Figure 22.

For *lbm*, as the number of 2D memory channels decreases, we see that the normalized execution time (with respect to Baseline policy suggested by Lo et al. [31]) for the TAM policy increases significantly. As 3D memory heats up, TAM migrates data from 3D to 2D memory without considering if the 2D memory is overloaded or slow. As 2D memory units decrease, it incurs significant overhead. EEFC and FC consider the loading on the 2D memory, and thus, are able to reduce overheads. EEFC is able to further improve upon the FC policy by selecting the channels to be closed at runtime.

6 EFFECT OF PROCESS VARIATIONS

Leakage power is a significant contributor to heating in 3D memories (Section 2). Leakage power consumption (and thus, temperature) is affected by the process variations occurring during

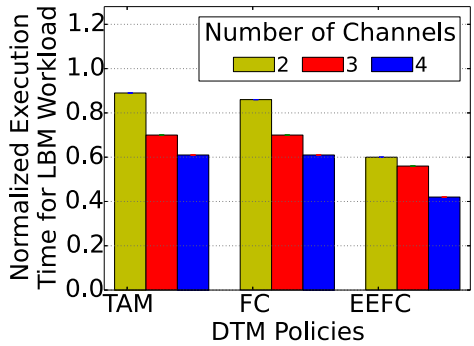


Fig. 22. Normalized execution time comparison with varying number of 2D memory channels.

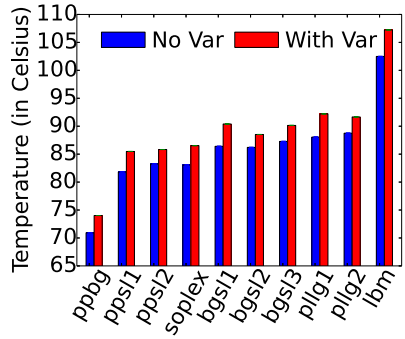


Fig. 23. Effect of variability on steady-state temperature of workloads.

manufacturing. Prior works have studied thermal issues due to process variation in 3D processors and addressed it by intelligent die stacking post-fabrication [21]. However, such approaches cannot adapt to change in workloads. Previous works have also studied the effect of process variation for 3D memories, but their focus has been limited to memory delay [6, 25, 48], latency [52], or power dissipation [7, 15]. None of these works address thermal issues in 3D memories. We study the effect of process variation on 3D memory temperatures and show that EEFC is relatively more resilient (lower application execution time and memory energy).

We motivate the study by comparing the steady-state temperature of the 3D memory when different application combinations are executed on a 16-core system (details in Section 6.2), without and with modeling of process variation (see Figure 23). We observe up to 5°C increase in the steady-state temperatures.

6.1 Variability Modeling

Variability, or process variation, affects the channel length and threshold voltage (V_t) of a transistor which, in turn, affects the leakage power of memory channels. Finally, leakage power, being a significant contributor to the overall power dissipation in 3D memories, affects the temperature of the 3D memory [43]. We obtain the amount of variations in leakage current from prior works and the variability distribution for different memory ranks using the VARIUS tool [42], which uses a multivariate Gaussian distribution in a spherical coordinate system to model spatially correlated process variations. It provides us with variability values for different regions of a die. We can specify the mean (μ) and standard deviation (σ) independently for die-to-die (D2D) and within-die

Table 6. Workloads and Benchmarks for Studying the Effect of Process Variations

Benchmark Details and Type	Workload Name
povray (×4), perlbench (×4), bwaves (×4), GemsFDTD (×4)	*ppbg*
povray (×4), perlbench (×4), soplex (×4), lbm (×4)	*ppsl* [1]
soplex (×16)	*soplex*
bwaves (×4), GemsFDTD (×4), soplex (×4), lbm (×4)	*bgsl* [1]
perlbench (×4), lbm (×8), GemsFDTD (×4)	*pllg* [1]
lbm (×16)	*lbm*

[1]Additional workloads are created by varying the application mapping to cores. A number is added as a suffix to denote this.

(WID) variations to obtain the variability maps. Since we consider 3D memories with eight layers, we generate eight variation maps of a die and stack them together to form a 3D memory device. The variation within a die is distributed as per μ_{WID} and σ_{WID} while the variation across dies follow μ_{D2D} and σ_{D2D}. For D2D leakage current variation, we use the nominal static current value as the mean (μ_{D2D}) while σ_{D2D} is reported as 9.2% in prior works [7].

In previous works [21], σ_{WID} value has been reported to be the same as σ_{D2D}. To ensure that the EEFC strategy works for a broad range of variability patterns, we show results in Section 6.3 for three different distributions: (a) W5D9 = {σ_{WID} = 5%, σ_{D2D} = 9.2%}, (b) W9D9 = {σ_{WID} = 9.2%, σ_{D2D} = 9.2%}, and (c) W9D5 = {σ_{WID} = 9.2%, σ_{D2D} = 5%}. We stack dies with higher mean leakage power closer to the heat sink as proposed by Juan et al. [21].

6.2 Experimental Setup

We evaluate the effect of variability using the same trace-based simulation setup discussed in Section 5.1, with modification to capture variation modeling.

Variation modeling. To model and quantify the effect of both D2D and WID variations on the leakage power/current consumption and the temperature, we generate variation maps from VARIUS [42] (Section 6.1). We generate variability patterns for 80 dies and report results for 10 different 3D memory devices (each with eight layers/dies) for each of the variability distributions. We scale HotSpot's leakage power calculation as per the variability patterns obtained from VARIUS. We also model thermally dependent leakage power in HotSpot, similar to [43].

Workloads. We use the same workloads as described in Section 5.1 to study the effect of process variation. Since all the compute-intensive benchmarks show similar behavior, we present results only for *ppbg*. We use identical workload mixes with benchmarks mapped differently (e.g., *bgsl1*, *bgsl2*, and *bgsl3*) to study the effect of application mapping onto different memory channels. The workloads are listed in increasing order of average memory access rates in Table 6.

6.3 Results and Discussion

We compare the performance results of EEFC with FC as the baseline policy. We analyze the normalized execution time and energy of EEFC with respect to FC for three different variability conditions; for each, we conduct experiments for 10 different 3D memory devices and show results using box plots, as illustrated in Figure 24. The workloads are shown in increasing order of memory intensiveness in all the plots.

6.3.1 Execution Time Comparison. Figure 25 shows the normalized execution time for W9D5 (σ_{WID} = 9.2%, σ_{D2D} = 5%). Execution time behavior for the other variability conditions (W9D9 and W5D9) are similar, and hence, are not shown.

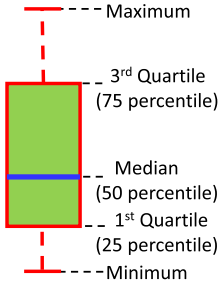


Fig. 24. Box plot elaboration.

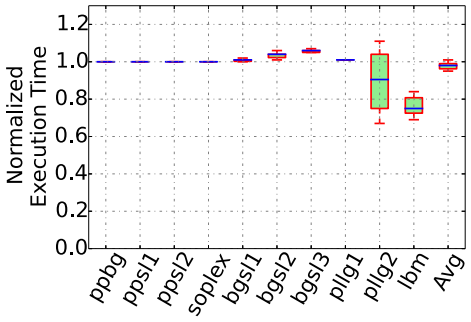


Fig. 25. Normalized execution time comparison for different workloads for $\sigma_{WID} = 9.2\%$, $\sigma_{D2D} = 5\%$.

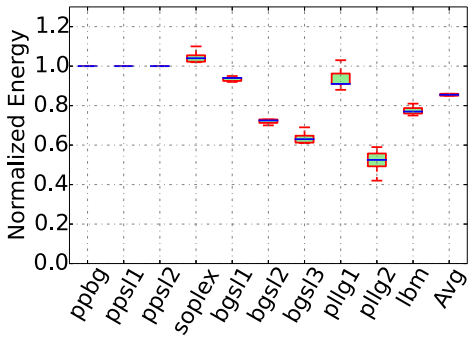
EEFC is able to comprehend variability and closes fewer channels than FC, thereby reducing the data migrations to/from 2D memory. It uses both temperature and leakage power as two independent parameters to decide on channel closing order, which allows the scalability of EEFC. In the presence of process variations, we might have channels with higher leakage but low access rate (also, temperature) and low leakage channels with high access rate/temperature. EEFC takes this into account in channel selection for closure, reducing the total number of closed channels. Across variabilities, we observe an average execution time improvement of 2%, with a maximum of 31% (Figure 25). Both 2D and 3D memories operate in parallel and the 2D memory bandwidth is sufficient to handle the migrated data accesses. Hence, from *ppbg* to *pllg1*, even though FC causes additional data transfers, it does not degrade the execution time. However, for *pllg2* and *lbm*, FC either closes more channels or closes memory-intensive channels, causing significant traffic on the 2D memory, thereby slowing it down and degrading the performance. Execution time slightly increases for *bgsl2, bgsl3, pllg1, pllg2* for certain devices, but energy and EDP reduce significantly (Figure 26(c)).

6.3.2 Temperature Comparison. We use the temperature-time trace of the *bgsl2* workload to visualize the DTM operation. As shown in Figure 27, both EEFC and FC follow similar trends. However, FC closes more memory channels (~6) and operates at a lower temperature than EEFC, which closes fewer channels (~4) and incurs lower overheads. Therefore, EEFC utilizes the temperature headroom more effectively.

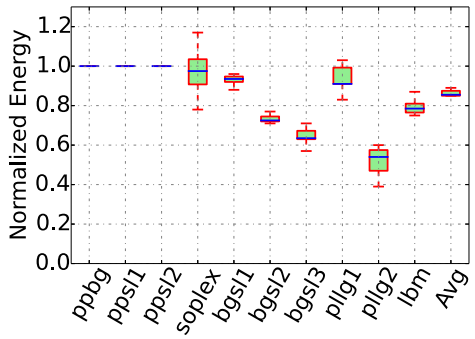
6.3.3 Energy Comparison. EEFC's flexibility in the order of closing channels results in lesser data movement between 3D and 2D memories, and keeps the 2D memory turned off for longer durations, leading to lower energy overheads (Figure 26).

Across variabilities, we observe an average improvement of 15% (up to 58%) in memory system energy consumption (3D + 2D). We also observe that EEFC is able to adapt to changes in application mapping and does not close memory-intensive channels, which reduces the memory energy significantly over FC such as in *bgsl2* and *bgsl3*, compared to *bgsl1*; and in *pllg2* compared to *pllg1*.

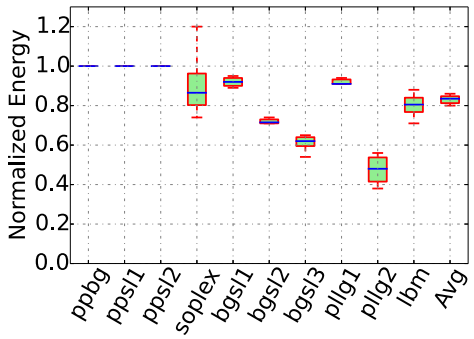
6.3.4 Impact of Variability. Using Figure 28, we analyze the effect of variability for *lbm*, a memory-intensive workload. EEFC results in 40% reduction in EDP across all variability conditions in comparison to FC. We observe that the EDP for EEFC has lower fluctuation with change in variability conditions, thereby indicating better adaptation to variability.



(a) σ_{WID} = 5%, σ_{D2D} = 9.2%



(b) σ_{WID} = 9.2%, σ_{D2D} = 9.2%



(c) σ_{WID} = 9.2%, σ_{D2D} = 5%

Fig. 26. Energy for different workloads using EEFC, for various variability conditions, normalized to corresponding energy for FC.

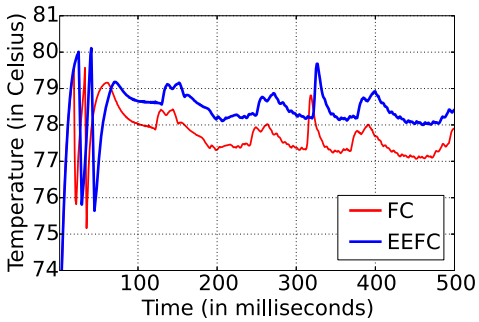


Fig. 27. Temperature-time trace for *bgsl2*.

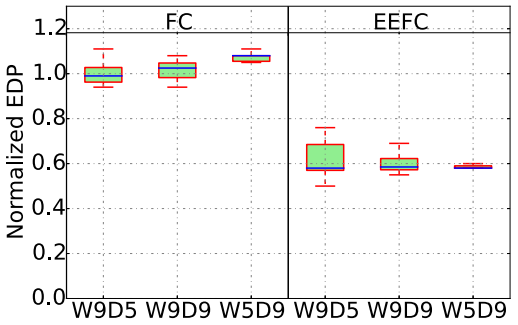


Fig. 28. EDP of FC and EEFC normalized to average EDP of FC (W9D5) for *lbm* workload.

6.4 Sensitivity Analysis

We discuss the sensitivity analysis of EEFC in the presence of process variations, highlighting the impact of certain key parameters on EDP.

Impact of considering access rate. Figure 29 shows the EDP for various workloads when access rate condition is considered for EEFC normalized to the case when it is not considered. We

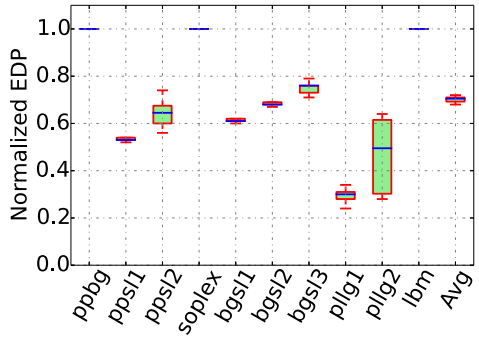


Fig. 29. Benefits of considering access rate for EEFC.

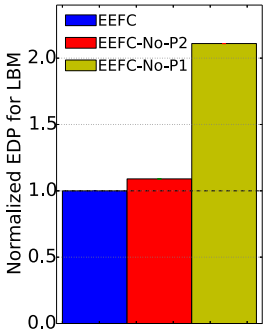


Fig. 30. Normalized EDP for LBM.

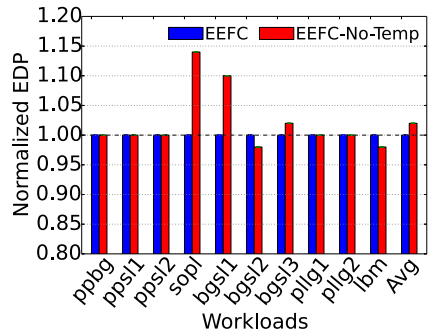


Fig. 31. Benefits of considering temperature for EEFC.

observe that considering access rate leads to an EDP reduction of up to 78% (30% on average across workloads) compared to not considering it within the EEFC strategy. The workloads *ppbg*, *soplex*, and *lbm* do not show any improvement because they have almost uniform access rates for all the channels and thus, every channel is classified as exhibiting relatively low access (since $ACC\_MULT > 1$). However, other benchmarks, where the access rate is not uniform across channels, show considerable EDP reduction.

Impact of Pass 1 and Pass 2. To justify the need for a three-pass strategy, we study the effect of bypassing Pass 1 or Pass 2 within the EEFC strategy. Pass 1 is beneficial for workloads with uniform access rate distribution across different channels (e.g., *lbm*). Pass 1 provides an average EDP improvement of 10% (up to 102%) and Pass 2 provides an additional EDP improvement of 1% (up to 9%), thereby motivating the importance of each pass.

Pass 1 and Pass 2 cause a limited change in execution time and EDP for compute-intensive and mixed workloads as these benchmarks migrate less amount of data. Memory-intensive workloads such as *lbm* show a considerable increase if either Pass 1 or Pass 2 is removed, as shown in Figure 30. For the *lbm* workload, the removal of Pass 1 and Pass 2 increases EDP to 2.12× and 1.09×, respectively.

Impact of considering temperature. As pointed out in Section 4.5.1, either temperature or leakage power could be used as a criterion for selecting the channels for closing. For Pass 2, our experiments showed that using temperature results in a 2% average EDP reduction compared to using leakage power and hence, we use temperature in Pass 2 (Figure 31).

7 CONCLUSION

We modeled the effect of leakage on 3D memory temperature and proposed a novel algorithm (FastCool) that turns off specific channels to maximize performance under thermal constraints. Data is migrated to 2D memory before closing a 3D memory channel. We developed an analytical model to quantify the 2D memory delay and used it to guide decisions. We evaluated the strategy using SPEC CPU2006 workloads running on a 3D + 2D memory platform, where our simulation results indicate that FC results in an improvement of 22%, 19%, and 32% on an average (up to 57%, 72%, and 82%) in application performance, memory energy, and EDP, respectively, over the state-of-the-art policies that reduce dynamic power alone.

We further studied the thermal behavior of 3D memories while deploying FC and proposed EEFC. EEFC selects channels for closure at runtime and considers the temperature, leakage power, channel access rate, and the position of channels for DTM and is a more complex strategy compared to FastCool. EEFC results in an additional improvement of up to 30%, 30%, and 51% in performance, memory energy, and EDP compared to FastCool. We demonstrated the adaptability and resilience of EEFC toward process variations using three different distributions of WID and D2D process variations and 10 different memory instances for each of these distributions.

In the future, we plan to evaluate rank-level approaches over the present channel-level approach for finer control over leakage power and further improvement in energy consumption of 3D memory-based systems.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their valuable feedback.

REFERENCES

[1] ANSYS. 2013. ANSYS Icepak User's Guide.

[2] Avram Bar-Cohen. 2014. *Encyclopedia of Thermal Packaging, Set 2: Thermal Packaging Tools*. World Scientific.

[3] David Brooks and Margaret Martonosi. 2001. Dynamic thermal management for high-performance microprocessors. In *Proceedings of the HPCA 7th International Symposium on High-Performance Computer Architecture*. IEEE, 171–182.

[4] Andrea Calimera, Karthik Duraisami, A. Sathanur, Prassanna Sithambaram, R. Iris Bahar, Alberto Macii, Enrico Macii, and Massimo Poncino. 2008. Thermal-aware design techniques for nanometer CMOS circuits. *Journal of Low Power Electronics (JOLPE)* 4, 3 (2008), 374–384.

[5] Trevor E. Carlson, Wim Heirman, Stijn Eyerman, Ibrahim Hur, and Lieven Eeckhout. 2014. An evaluation of high-level mechanistic core models. *ACM Transactions on Architecture and Code Optimization (TACO)*, (2014), Article 5, 23 pages. DOI: https://doi.org/10.1145/2629677

[6] Karthik Chandrasekar, Sven Goossens, Christian Weis, Martijn Koedam, Benny Akesson, Norbert Wehn, and Kees Goossens. 2014. Exploiting expendable process-margins in DRAMs for run-time performance optimization. In *Design, Automation & Test in Europe Conference (DATE'14)*. European Design and Automation Association, 173.

[7] Karthik Chandrasekar, Christian Weis, Benny Akesson, Norbert Wehn, and Kees Goossens. 2013. Towards variation-aware system-level power estimation of DRAMs: An empirical approach. In *Design Automation Conference (DAC'13)*. ACM, 23.

[8] Ke Chen, Sheng Li, Naveen Muralimanohar, Jung Ho Ahn, Jay B. Brockman, and Norman P. Jouppi. 2012. CACTI-3DD: Architecture-level modeling for 3D die-stacked DRAM main memory. In *Design, Automation & Test in Europe Conference (DATE'12)*. EDA Consortium, 33–38.

[9] Hybrid Memory Cube Consortium. 2015. HMC Specification 2.1.

[10] Ayse Kivilcim Coskun, Tajana Simunic Rosing, and Kenny C. Gross. 2008. Proactive temperature management in MPSoCs. In *International Symposium on Low Power Electronics and Design (ISLPED'08)*. ACM, 165–170.

[11] Perceval Coudrain, Papa Momar Souare, Rafael Prieto, Vincent Fiori, Alexis Farcy, Laurent Le Pailleur, Jean-Philippe Colonna, Cristiano Santos, Pascal Vivet, Haykel Ben-Jamaa, et al. 2016. Experimental insights into thermal dissipation in TSV-Based 3D integrated circuits. *Design & Test*1 (2016), 1–1.

[12] David Cuesta, José L. Risco-Martín, José L. Ayala, and J. Ignacio Hidalgo. 2015. Thermal-aware floorplanner for 3D IC, including TSVs, liquid microchannels and thermal domains optimization. *Applied Soft Computing* 34, C (Sept. 2015), 164–177.

[13] Kapil Dev, Gary Woods, and Sherief Reda. 2013. High-throughput TSV testing and characterization for 3D integration using thermal mapping. In *2013 50th ACM/EDAC/IEEE Design Automation Conference (DAC'13)*. IEEE, 1–6.

[14] Karthik Elangovan, Ivan Rodero, Manish Parashar, Francesc Guim, and Isaac Hernandez. 2011. Adaptive memory power management techniques for HPC workloads. In *Proceedings of the 2011 18th International Conference on High Performance Computing*. IEEE, 1–11.

[15] Saugata Ghose, Abdullah Giray Yaglikçi, Raghav Gupta, Donghyuk Lee, Kais Kudrolli, William X. Liu, Hasan Hassan, Kevin K. Chang, Niladrish Chatterjee, Aditya Agrawal, et al. 2018. What your DRAM power models are not telling you: Lessons from a detailed experimental study. *ACM on Measurement and Analysis of Computing Systems (SIGMETRICS)* 2, 3 (2018), 38.

[16] Mohammad Hossein Hajkazemi, Mohammad Khavari Tavana, Tinoosh Mohsenin, and Houman Homayoun. 2017. Heterogeneous HMC+DDRx memory management for performance-temperature tradeoffs. *ACM Journal on Emerging Technologies in Computing Systems (JETC)* 14, 1, (Sept. 2017), Article 4, 21 pages. DOI: https://doi.org/10.1145/3106233

[17] F. Hameed, M. A. A. Faruque, and J. Henkel. 2011. Dynamic thermal management in 3D multi-core architecture through run-time adaptation. In *Design, Automation & Test in Europe Conference (DATE'11)*. 1–6. DOI: https://doi.org/10.1109/DATE.2011.5763053

[18] Hai Huang, Kang G. Shin, Charles Lefurgy, and Tom Keller. 2005. Improving energy efficiency by making DRAM less randomly accessed. In *International Symposium on Low Power Electronics and Design (ISLPED'05)*. ACM, 393–398.

[19] Bruce Jacob. 2009. The memory system: You can't avoid it, you can't ignore it, you can't fake it. *Synthesis Lectures on Computer Architecture* 4, 1 (2009), 1–77.

[20] Joe Jeddeloh and Brent Keeth. 2012. Hybrid memory cube new DRAM architecture increases density and performance. In *Symposium on VLSI Technology (VLSIT'12)*. IEEE, 87–88.

[21] Da-Cheng Juan, Siddharth Garg, and Diana Marculescu. 2014. Statistical peak temperature prediction and thermal yield improvement for 3D chip multiprocessors. *ACM Transactions on Design Automation of Electronic Systems (TODAES)* 19, 4 (2014), 39.

[22] Dhireesha Kudithipudi, Qinru Qu, and Ayse K. Coskun. 2013. Thermal management in many core systems. In *Evolutionary Based Solutions for Green Computing*. Springer, 161–185.

[23] Sumeet S. Kumar, Amir Zjajo, and Rene van Leuken. 2017. Fighting dark silicon: Toward realizing efficient thermal-aware 3-D stacked multiprocessors. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 25, 4 (2017), 1549–1562.

[24] Benoit Lasbouygues, Robin Wilson, Nadine Azemard, and Philippe Maurine. 2007. Temperature-and voltage-aware timing analysis. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)* 26, 4 (2007), 801–815.

[25] Hyun-Woo Lee, Ki-Han Kim, Young-Kyoung Choi, Ju-Hwan Sohn, Nak-Kyu Park, Kwan-Weon Kim, Chulwoo Kim, Young-Jung Choi, and Byong-Tae Chung. 2011. A 1.6V 1.4 Gbp/s/pin consumer DRAM with self-dynamic voltage scaling technique in 44nm CMOS technology. *IEEE Journal of Solid-State Circuits (JSSC)* 47, 1 (2011), 131–140.

[26] Chien-Hui Liao, Charles H-P. Wen, and Krishnendu Chakrabarty. 2015. An online thermal-constrained task scheduler for 3D multi-core processors. In *Design, Automation & Test in Europe Conference (DATE'15)*. 351–356.

[27] Weiping Liao, Lei He, and Kevin Lepak. 2004. *Temperature-Aware Performance and Power M*. Technical Report 04-250. UCLA Engr. Citeseer.

[28] Weiping Liao, Lei He, and Kevin M. Lepak. 2005. Temperature and supply voltage aware performance and power modeling at microarchitecture level. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 24, 7 (2005), 1042–1053.

[29] Ankur Limaye and Tosiron Adegbija. 2018. A workload characterization of the SPEC CPU2017 benchmark suite. In *International Symposium on Performance Analysis of Systems and Software (ISPASS'18)*. 149–158. DOI: https://doi.org/10.1109/ISPASS.2018.00028

[30] Shaobo Liu, Jingyi Zhang, Qing Wu, and Qinru Qiu. 2010. Thermal-aware job allocation and scheduling for three dimensional chip multiprocessor. In *Proceedings of ISQED'10*. 390–398. DOI: https://doi.org/10.1109/ISQED.2010.5450547

[31] Wei-Hen Lo, Kai-zen Liang, and TingTing Hwang. 2016. Thermal-aware dynamic page allocation policy by future access patterns for Hybrid Memory Cube (HMC). In *Design, Automation & Test in Europe Conference (DATE'16)*. 1084–1089.

[32] Gian Luca Loi, Banit Agrawal, Navin Srivastava, Sheng-Chih Lin, Timothy Sherwood, and Kaustav Banerjee. 2006. A thermally-aware performance analysis of vertically integrated (3D) processor-memory hierarchy. In *Design Automation Conference (DAC'06)*. 991–996.

[33] Yanchao Lu, Donghong Wu, Bingsheng He, Xueyan Tang, Jianliang Xu, and Minyi Guo. 2016. Rank-aware dynamic migrations and adaptive demotions for DRAM power management. *IEEE Transactions on Computers* 65, 1 (Jan. 2016), 187–202. DOI: https://doi.org/10.1109/TC.2015.2409847

[34] Jie Meng and Ayse K. Coskun. 2012. Analysis and runtime management of 3D systems with stacked DRAM for boosting energy efficiency. In *Design, Automation & Test in Europe Conference (DATE'12)*. 611–616. DOI: https://doi.org/10.1109/DATE.2012.6176545

[35] Jie Meng, Katsutoshi Kawakami, and Ayse K. Coskun. 2012. Optimizing energy efficiency of 3-D multicore systems with stacked DRAM under power and thermal constraints. In *Design Automation Conference (DAC'12)*. IEEE, 648–655.

[36] Sai Prashanth Muralidhara, Lavanya Subramanian, Onur Mutlu, Mahmut Kandemir, and Thomas Moscibroda. 2011. Reducing memory interference in multicore systems via application-aware memory channel partitioning. In *IEEE/ACM International Symposium on Microarchitecture (MICRO'11)*. ACM, 374–385.

[37] Santiago Pagani, Heba Khdr, Jian-Jia Chen, Muhammad Shafique, Minming Li, and Jörg Henkel. 2014. TSP: Thermal safe power: Efficient power budgeting for many-core systems in dark silicon. In *International Conference on Hardware/Software Codesign and System Synthesis*. ACM, 10.

[38] Vasilis F. Pavlidis and Eby G. Friedman. 2009. *Three-Dimensional Integrated Circuit Design*. Morgan Kaufmann Publishers Inc., San Francisco, CA.

[39] Ivy Bo Peng, Roberto Gioiosa, Gokcen Kestor, Pietro Cicotti, Erwin Laure, and Stefano Markidis. 2017. Exploring the performance benefit of hybrid memory system on HPC environments. In *2017 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW'17)*. IEEE, 683–692.

[40] Zhiliang Qian and Chi-Ying Tsui. 2011. A thermal-aware application specific routing algorithm for network-on-chip design. In *Asia and South Pacific Design Automation Conference (ASPDAC'11)*. IEEE, 449–454.

[41] Mohamed M. Sabry, Ayse K. Coskun, David Atienza, Tajana Šimunić Rosing, and Thomas Brunschwiler. 2011. Energy-efficient multiobjective thermal control for liquid-cooled 3-D stacked architectures. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)* 30, 12 (2011), 1883–1896.

[42] Smruti R. Sarangi, Brian Greskamp, Radu Teodorescu, Jun Nakano, Abhishek Tiwari, and Josep Torrellas. 2008. VARIUS: A model of process variation and resulting timing errors for microarchitects. *IEEE Transactions on Semiconductor Manufacturing (TSM)* 21, 1 (2008), 3–13.

[43] Lokesh Siddhu and Preeti Ranjan Panda. 2019. FastCool: Leakage aware dynamic thermal management of 3D memories. In *Design, Automation & Test in Europe Conference (DATE'19)*. IEEE, 272–275.

[44] Lokesh Siddhu and Preeti Ranjan Panda. 2019. PredictNcool: Leakage aware thermal management for 3D memories using a lightweight temperature predictor. *ACM Transactions on Embedded Computing Systems (TECS)* 18, 5s (2019), 64.

[45] Filippo Sironi, Martina Maggio, Riccardo Cattaneo, Giovanni F. Del Nero, Donatella Sciuto, and Marco D. Santambrogio. 2013. ThermOS: System support for dynamic thermal management of chip multi-processors. In *PACT*. IEEE Press, 41–50.

[46] Avinash Sodani. 2015. Knights landing (KNL): 2nd generation Intel® Xeon Phi processor. In *2015 IEEE Hot Chips 27 Symposium (HCS)*. IEEE, 1–24.

[47] Sadagopan Srinivasan, Li Zhao, Brinda Ganesh, Bruce Jacob, Mike Espig, and Ravi Iyer. 2009. CMP memory modeling: How much does accuracy matter. In *Modeling, Benchmarking and Simulation (MoBS)*.

[48] Meysam Taassori, Ali Shafiee, and Rajeev Balasubramonian. 2016. Understanding and alleviating intra-die and intra-DIMM parameter variation in the memory system. In *International Conference on Computer Design (ICCD'16)*. IEEE, 217–224.

[49] Yuan Xie, Jason Cong, and Sachin Sapatnekar. [n.d.]. Three-dimensional integrated circuit design. ([n.d.]).

[50] Marina Zapater, Jose L. Ayala, José M. Moya, Kalyan Vaidyanathan, Kenny Gross, and Ayse K. Coskun. 2013. Leakage and temperature aware server control for improving energy efficiency in data centers. In *Design, Automation & Test in Europe Conference (DATE'13)*. 266–269. DOI: https://doi.org/10.7873/DATE.2013.067

[51] Runjie Zhang, Mircea R. Stan, and Kevin Skadron. 2015. *HotSpot 6.0: Validation, Acceleration and Extension*. Technical Report. University of Virginia.

[52] Bo Zhao, Yu Du, Youtao Zhang, and Jun Yang. 2009. Variation-tolerant non-uniform 3D cache management in die stacked multicore processor. In *IEEE/ACM International Symposium on Microarchitecture (MICRO'09)*. ACM, 222–231.

[53] Jintao Zheng, Ning Wu, Lei Zhou, Yunfei Ye, and Ke Sun. 2016. DFSB-based thermal management scheme for 3D NoC-bus architectures. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 24, 3 (March 2016), 920–931. DOI: https://doi.org/10.1109/TVLSI.2015.2439698

Received March 2020; revised July 2020 and August 2020; accepted August 2020