3D-TemPo: Optimizing 3D DRAM Performance Under Temperature and Power Constraints

ABSTRACT

3D DRAM provides a significant performance boost resulting from enormous memory bandwidth. However, the stacked memory architecture exhibits high power density causing thermal hotspots. Further, systems under power constraints require careful planning for intelligent allocation of the available power to its various components. A straightforward dynamic power management policy of allocating more power to potentially high memory activity 3D DRAM ranks to maximize system performance causes a rise in the temperature of such ranks, making them susceptible to thermal stalls and shutdown by dynamic thermal management (DTM) strategies. A rise in rank temperature, in turn, increases the leakage power of memory ranks, affecting power budgeting decisions. Thus, coordinated power budgeting and thermal management are needed. We propose an adjacency-aware dynamic power budgeting technique, *3D-TemPo*, which dynamically performs a reward-based power allocation to memory ranks, and leverages strong thermal correlations between vertically adjacent ranks to maximize 3D DRAM performance under power and thermal constraints. We evaluate *3D-TemPo* using SPEC CPU2017, and SPLASH2 workloads and observe average execution time improvement of 24% compared to baseline strategies.

1 INTRODUCTION

Emerging memory technologies such as 3D-stacked DRAMs [1, 2], offering enormous memory bandwidth (as high as 386 GBps), are being commercialised in an attempt to break the *memory wall*. High Bandwidth Memory (HBM) [1] has been deployed in modern GPUs and AI processors [3] to meet the bandwidth requirements of memory-intensive workloads involving huge data movement between processing cores and DRAM. 3D stacking enables hundreds of banks/ ranks per memory device, however, the off-chip memory bandwidth is often limited due to the power budget allocated for DRAM [15–17]. Vertical integration results in high power density and poor heat dissipation capability, causing thermal challenges. Systems running under a power budget and a thermal constraint employ dynamic power budgeting and dynamic thermal management (DTM) policies to ensure reliability. The performance of 3D DRAM relies heavily on these policies that enable a subset of ranks based on power and thermal constraints. Such policies affect each other and can not be performed in isolation. The static/leakage power of memory shows an exponential rise with increasing temperature, motivating for the need of a thermal-aware power budgeting policy. Similarly, the temperature profile of memory ranks depend on the power budgeting decisions; the heating occurs from memory activity permitted by the power budgeting policy.

3D DRAM consists of several vertically stacked DRAM dies with heterogeneous power and thermal status. The top dies, located closer to the heat sink, exhibit better heat dissipation; the bottom dies, located away from the heat sink, exhibit higher temperatures even when their memory activity (and therefore their dynamic power consumption) is low. Each DRAM die consists of multiple physical channels, each channel composed of several ranks. 3D DRAM exhibits a strong thermal co-relation between vertically adjacent DRAM ranks/channels compared to those within the same die. The heterogeneity in 3D DRAM leading to higher temperatures at lower power dissipation and vice-versa due to stacked architecture, motivates us to propose an adjacency-aware dynamic power budgeting policy. We make following important observations: (1) the physical location of a channel in the stacked architecture is crucial in determining the *progress* obtained by enabling the channel, (2) allocating power budget to high memory activity channels does not always guarantee optimal performance as they are more prone to heating (due to high dynamic power dissipation) leading to thermal shutdown, thereby stalling the cores, and (3) enabling vertically adjacent high memory activity channels contributes to more loss than gain as vertically aligned thermal hotspots drastically reduce 3D DRAM's cooling efficiency and increase the thermal stall durations.

In this work, we make the following specific contributions:

(1) We propose a thermal-aware dynamic power budgeting policy, *TemPo*, which periodically suggests the ideal set of channels that should be enabled with a given power budget and under a thermal constraint. This is the first work, to the best of our knowledge, to propose coordinated thermal and power management to maximize 3D DRAM performance.
(2) We consider a non-uniform traffic amongst memory channels, which results in varying thermal profile across the 3D stack, making power budgeting decisions non-trivial.

2 RELATED WORK

With increasing power density in novel memory technologies, power budgeting and thermal analysis of 3D architectures becomes important [19–21]. Prior works have proposed solutions to efficient power budgeting and thermal management for homogeneous and heterogeneous multi-core systems using task migration and DVFS [4], workload's power profile [5], QoS-aware frequency throttling [6], temperature prediction [7, 10], and transient temperature aware budgeting [8]. For 3D multi-core architecture (multiple layers of cores stacked together), Coskun et al. [9] proposed coordinated power and thermal management using job scheduling and DVFS, and Zhu et al. [12] proposed a power-thermal budgeting solution leveraging the heterogeneous thermal characteristics of processing cores in different layers. Meng et al. [11] used DVFS on cores to optimize energy efficiency in 3D multi-core system with stacked DRAM, assuming uniform traffic across all DRAM banks. Prior works have targetted power budgeting, alongside thermal management, primarily for the processing cores. We target the power budgeting problem under thermal constraint for 3D DRAM, assuming different subsets of cores mapped to different memory channels, thereby producing non-uniform traffic across different 3D DRAM banks depending on the workload runtime behavior.

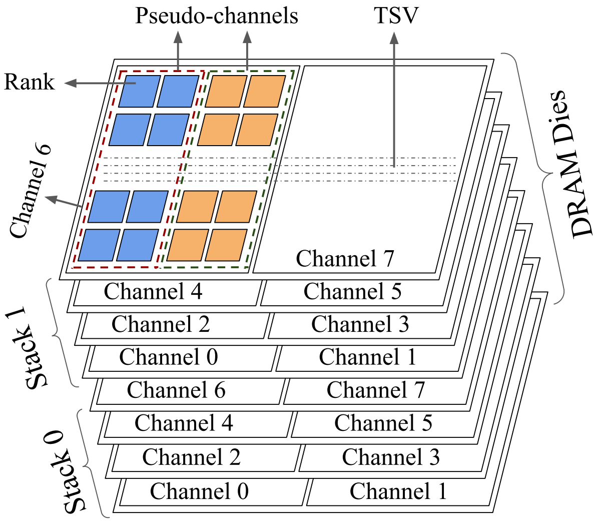


Figure 1: High Bandwidth Memory (HBM2E) structure

3 DYNAMIC POWER BUDGETING AND THERMAL MANAGEMENT

3.1 Memory Architecture

A 3D DRAM consists of multiple stacks, with each stack comprising several DRAM dies stacked vertically. DRAM dies are connected using fast through-silicon via (TSV) based interconnects. A portion of the DRAM die from each stack is connected to the same independent physical channel, each channel consisting of two pseudo-channels. Each pseudo-channel contains multiple ranks, which in turn contain banks which are organised into rows and columns, similar to a conventional 2D DRAM. Figure 1 shows the structure of 3D DRAM having 2 stacks (0 and 1), 4 DRAM dies and 8 channels per stack, 2 pseudo-channels per channel, and 8 ranks per pseudo-channel. Each channel (0 to 7) services a total of 16 banks from two DRAM dies located in stack 0 & 1. A subset of cores are mapped to each of these channels; thus, the memory traffic generated by each core is restricted to its corresponding channel only.

Modern DRAMs support multiple power states that are usually controlled at a rank level. Different power states differ in their power consumption with *read/write* state consuming maximum power (dynamic + leakage). The *standby* state, which does not permit memory accesses, consumes a fraction of leakage power required only to retain the data. In the *active* state, the memory consumes higher leakage power, and is prepared for accesses. The power state transition of ranks requires a minimal overhead (typically of the order of a few ns to a few μs) [13]. Our dynamic power budgeting policy allocates power at the level of a memory channel for a given power budget. A channel is activated if all of its constituent ranks are either in *active* or *read/write* state. Similarly, to deactivate a channel, the budgeting technique sets all its ranks to *standby* state.

3.2 Problem Definition

We consider a performance optimization problem on 3D DRAM system running under power and thermal constraints. Given: (1) a multi-core processor with p processing cores, (2) a 3D DRAM memory with c channels and r ranks, each channel mapped to a set of cores, and (3) a memory power budget P_b, and temperature constraint T_{crit}, we aim to minimize the workload's execution time ET, subject to:

$$T_{\mathrm{i}} \leq T_{\mathrm{crit}}, i \in \{0, 1, \ldots, (r-1)\}, \textit{and} \\ \sum_{i \in \{0,1,\ldots,(r-1)\}} P_i \leq P_b \tag{1}$$

where T_i and P_i denote the temperature and the power consumption of rank i.

3.3 Round Robin Power Budgeting

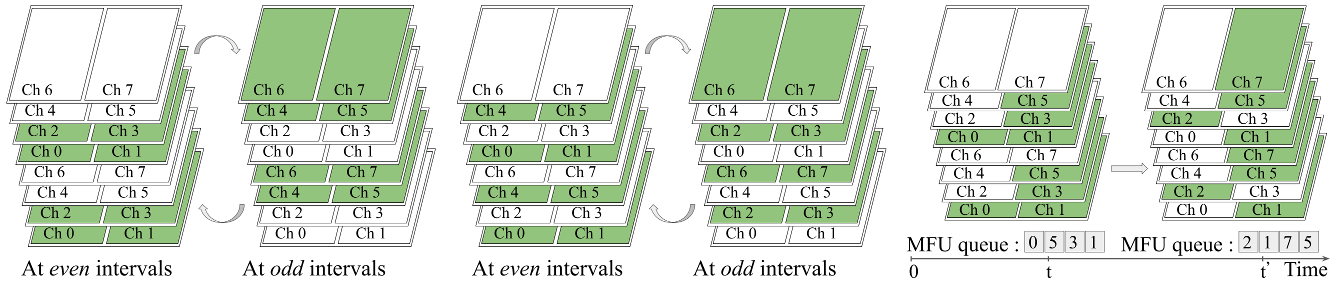
Figure 2a illustrates the working of round robin power budgeting policy. The round robin policy activates Channels 0 to 3 in the even intervals, and activates Channels 4 to 7 in the odd intervals of workload execution, with an interval (or epoch) being defined as the duration between two consecutive invocations of the policy. This policy attempts to ensure fairness amongst all the memory channels and therefore amongst cores mapped to the channels. However, such a policy treats all the cores uniformly and often leads to sub-optimal performance as the cores running *compute-heavy* tasks might get starved. Furthermore, as stated earlier, 3D DRAM exhibits a strong thermal correlation between vertically adjacent channels/ ranks. Due to this, the round robin based activation order leads to more frequent thermal stalls, causing performance penalty.

3.4 Alternation Power Budgeting

Figure 2b shows the working of *alternation* policy that attempts to eliminate the vertically aligned thermal hotspots in 3D DRAM. The policy allocates power to channels in the alternate DRAM dies in both even and odd intervals. While this helps to improve DRAM cooling and reduces thermal stall durations, the starvation of *compute-heavy* cores still remains unresolved, affecting system performance. The policy also ensures fairness amongst cores; however, it often leads to sub-optimal performance as the memory activity rate in channels is not taken into consideration, leading to uniform treatment of highly accessed and less accessed channels.

3.5 Most Frequently Used Power Budgeting

The memory activity rate across different channels is an important indicator towards achieving maximum 3D DRAM utilization. Figure 2c shows the working of *most frequently used* (MFU) policy that maintains a MFU queue of channels for its operation. It allocates the available power budget to the most frequently used channels of the previous interval, leading to high memory throughput. However, for thermally-constrained systems, a sharp rise in frequently used channels' temperatures is observed due to continuous activation. This in turn increases the number of thermal stalls due to channel's high dynamic and leakage power dissipation. At each time interval, the channels at the head of MFU queue are activated while the power budget permits. The policy ignores the accumulation of vertically aligned thermal hotspots and severely stalls the *compute-heavy* cores as the channels mapped to them are least preferred for activation. It only considers optimizing the immediate memory throughput overlooking the associated thermal impact.



(a) Round Robin **(b) Alternation** **(c) Most Frequently Used**

Figure 2: Working of power budgeting policies. *Shading* represents *activated* channels.

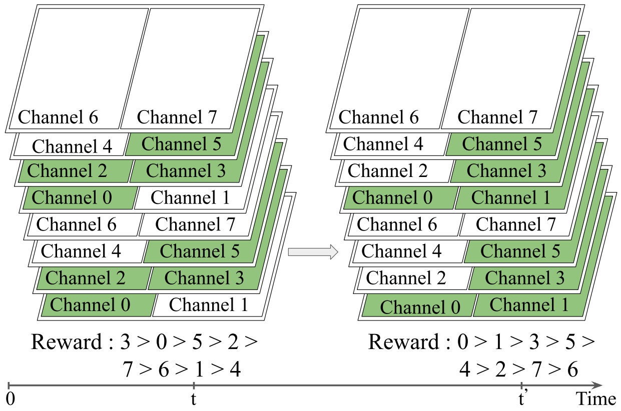


Figure 3: Working of *Reward-based* policy

3.6 Reward-based Power Budgeting

We propose a *reward-based* power budgeting policy that jointly considers the channel's memory activity and the *core progress* obtained through activating that channel. Power budgeting is formulated as an instance of the classical Knapsack problem problem as follows:

(1) The total 3D DRAM power budget P_b forms the *capacity* of the knapsack.
(2) The memory channels constitute the *objects*.
(3) The *weight* of memory channels is derived from three power components per rank per interval: (1) dynamic power (P_{dyn}) proportional to the memory access counts in the *read/write* state, (2) temperature-dependent leakage power (P_{leak}), and (3) average refresh power (P_{ref}). The weight of a memory channel is sum of the power consumption of all its ranks.
(4) The *profit* obtained by activating a channel (similar to including an object in knapsack) is computed as the sum of the *progress* made by the corresponding cores mapped to this channel. We measure progress of a core as its instructions-per-cycle (IPC) in the last interval.

Assuming a system with p cores, c channels, and r ranks with p/c cores mapped to each channel and r/c ranks in each channel, we calculate the *reward* of activating a channel ch in each interval as the ratio of its *profit* to its *weight* as shown in Equation 2. The various power components of a memory rank j are computed online as follows: (1) $P_{dyn}(j)$ is computed as the product of memory access counts of j and energy per read/write access, divided by the time interval, (2) $P_{leak}(j)$ is obtained through a lookup table (Section 4.5, and (3) $P_{ref}(j)$ is taken to be a constant.

$$Reward_{ch} = \frac{\sum\limits_{i \in map(ch)} IPC(i)}{\sum\limits_{j \in ch} (P_{dyn}(j) + P_{leak}(j) + P_{ref}(j))} \tag{2}$$

The proposed *reward-based* policy sorts the channel *rewards* in non-increasing order and activates channels in that order, as shown in Figure 3. Unlike *round robin*, the *reward-based* policy does not starve the *compute-heavy* cores which do not make frequent memory requests. Rather, it prioritizes such cores, ensuring maximum system progress with minimum memory power dissipation. Unlike MFU, this policy prioritizes the channels based on their role in making system progress and not merely on the instantaneous memory throughput. The *reward-based* strategy acknowledges a channel's physical location in 3D DRAM by considering the temperature-dependent leakage power. Top channels undergo a slow rise in temperature, consuming less leakage power compared to the bottom channels even when subjected to similar memory activity rates. This helps in making better power budgeting decisions by prioritizing top channels over bottom ones when similar amount of *progress* is expected in the corresponding cores, reducing the thermal stalls.

3.7 Adjacency-aware 3D-TemPo Power Budgeting

We identify three regions representing the *thermal state* of 3D DRAM during the workload execution, based on the maximum current memory temperature (T_{max}): (1) the *cool* region with $T_{max} < T_{cool}$, (2) the *hot* region with $T_{cool} \leq T_{max} < T_{hot}$, and (3) the *critical* region with $T_{hot} \leq T_{max} \leq T_{crit}$. In the *cool* region, our *3D-TemPo* policy performs power allocation based on memory activity rate of channels. As the thermal emergency does not occur in this region, prioritizing higher access frequency channels is most beneficial. In the *hot region*, the channels are susceptible to thermal stalls so our policy uses the *reward-based* budgeting which considers both IPC of the cores and the DRAM's thermal condition and helps to defer the thermal stalls. To prevent vertically aligned

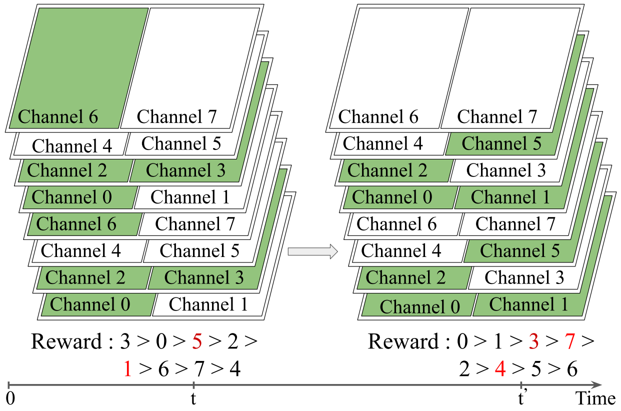


Figure 4: Working of *Adjacency-aware 3D-TemPo* policy

thermal hotspots overlooked by *reward-based* policy, we leverage adjacency-awareness in *3D-TemPo* policy for the *critical* region of 3D DRAM. The adjacency-awareness additionally exploits the observed thermal correlation between vertically adjacent channels. Figure 4 shows the working of the adjacency-aware *3D-TemPo* policy. While allocating power to the channels in the order of high channel reward, the policy skips the channels that are vertically adjacent to any of the activated channels. This prevents the trapping of heat between vertically adjacent channels, minimizing DTM penalty. We only skip a high reward channel when its vertically adjacent neighbours n are in the *critical region* ($T_{hot} \leq T_n \leq T_{crit}$). For example, channels 5 in Figure 4 is skipped as channel 3 is in the *critical region* ($T_{ch3} > T_{hot}$). However, channel 2, vertically adjacent to activated channel 0, is not skipped as $T_{ch0} < T_{hot}$.

3.8 DRAM Low-power Based DTM

As discussed earlier, modern DRAM allows rank-level control on the power states and helps to manage power dissipation. We employ the DRAM power states to also regulate the memory channel temperatures and maintain the temperature constraint. Our DTM policy performs the management at the channel level, whereby, a channel is sent to low-power *standby* state (causing thermal stall) if any of its ranks exceeds T_{crit}. Similarly, a channel is reverted to *read/write* state upon cooldown of all its ranks. During thermal emergencies in a channel, the power budgeting policy does not consider it for power allocation irrespective of the associated reward or priority. The DTM policy sends the temperature of memory channels to the budgeting policy at the beginning of each interval.

3.9 The Overall Flow

The *3D-TemPo* policy (Algorithm 1) uses low-power states based DTM for ensuring safe thermal limits and performs reward-based and adjacency-aware power allocation to memory channels. The workload execution is divided into time intervals (epochs), and at the beginning of every epoch, the maximum channel temperature T_{max} is obtained (Line 1). If one or more channels have exceeded $\mathrm{T_{crit}}$ (temperature constraint), the DTM policy is invoked and appropriate rank states (*RankState*) are computed (Lines 2-4). For each potentially available channels, not yet set to *standby* state by DTM, the channels are activated in order of high memory activity in the *cool* region (Lines 5-6). In the *hot* or *critical* region, the channel reward is computed and sorted in non-increasing order (Lines 8-10). Based on the available power budget P_b, the top reward channels are selected, checked for *critical* vertically adjacent neighbours (temperature ≥ T_{hot}), activated upon *non-critical* vertical neighbours, and skipped otherwise, updating the power budget (Lines 11-17). Upon cool down of all channel ranks below T_{rec}, the channel is reset to *active* state from *standby* (Line 18). Finally, the rank states and the activated channels are returned for the current epoch (Line 19) and sent to the memory controller for appropriate action.

Algorithm 1: Adjacency-aware 3D-TemPo Power Budgeting

```
   Input: $P_b$: Memory power budget
   Input: $T[0:(c-1)]$: Memory channel temperatures
   Input: $T_{crit}$: DTM invocation temperature
   Input: $T_{rec}$: Recovery temperature
   Input: $T_{cool}$: Threshold temperature for cool region
   Input: $T_{hot}$: Threshold temperature for vertical neighbours
   Output: $RankState[0:(r-1)]$: Power state of ranks,
           $Activated[0:(c-1)]$: Channel status
1  $T_{max} \leftarrow$ Get_Max_Channel_Temperature ($T$)
2  if $T_{max} > T_{crit}$ then // Heated. Apply DTM.
3      Invoke_DTM_Policy ()
       // Place hot channels in standby.
4      $RankState \leftarrow$ Rank_Power_State ($T$,$T_{crit}$)
5  else if $T_{max} < T_{cool}$ then // Cool region.
6      Activate_High_Activity_Channels ($P_b$)
7  else // Hot or critical region.
8      for each channel ch in {0, 1, ... , (c-1)} do
           // Compute channel reward using Eq. 2
9          $Reward[ch] \leftarrow$ Compute_Reward ( )
10     $TopRewardChannels \leftarrow$ Sort_Rewards ($Reward$)
11     for each channel ch in $TopRewardChannels$ do
           // Check if vertically adjacent neighbours
              in critical region.
12         $Adj \leftarrow$ Is_Vertical_Neighbour_Critical ($ch$, $T_{hot}$)
           // Non-critical vertical neighbours.
13         if $Adj == False$ then
14             $P_{ch} \leftarrow$ Compute_Req_Power ($ch$)
15             if $P_b \geq P_{ch}$ then
                   // Activate the channel.
16                 Activate_Channel ($ch$)
                   // Update Power Budget.
17                 $P_b$ -= $P_{ch}$
   // Place recovered channels in read/write state
18 $RankState \leftarrow$ Set_Active_On_Recovery ($T$,$T_{rec}$)
19 return $RankState$, $Activated$
```

4 EXPERIMENTAL EVALUATION

4.1 Simulation Environment

We use an integrated performance-thermal simulator CoMeT [14], consisting of Sniper 7.2 multicore simulator and Hotspot 6.0, running the workload, collecting memory access counts, and performing power allocation to channels every 1 ms (epoch time, *E*). We obtain the refresh and dynamic power dissipation using energy-per-access values (24.45 nJ per 64-byte access) from CACTI-3DD and feed the power values to HotSpot thermal simulator with default configuration parameters [18]. Computed temperatures are sent back to Sniper, where the dynamic power budgeting and thermal management decisions are implemented.

We model a multi-core processor (64 cores, 3.6GHz, 22 nm, out-of-order, caches: 32KB private L1, 256 KB private L2, 8 MB shared L3) with an off-chip 3D DRAM (8GB size, 8 channels, 2 pseudo-channels per channel, 16 ranks/pseudo-channel, 1 bank/rank, 29 ns latency, and 44 GBps per channel bandwidth), and running workloads comprising 64 applications, one on each core. The *standby* state consumes 17% static power and the transition overhead to *active* state is $\sim 6\mu s$ [13], negligible compared to our epoch time. We empirically determine the temperature thresholds: T_{rec}=77°C, T_{crit}=80°C (similar to [18]), T_{cool}=74°C, and T_{hot}=78°C.

We use workloads from SPLASH-2 and SPEC CPU2017 benchmark suites to validate the efficacy of our proposal. Table 1 lists the selected workloads from each suite. We simulate the compiled source code for SPLASH2 workloads and pre-generated traces (Pinballs) for 100M instructions for SPEC CPU2017 benchmarks.

4.2 Performance Improvement

Figure 5 and 6 show the execution time (normalized to *NoCons*) of different policies for two different power budgets, 50W and 25W, accounting for 50% and 25% of peak power consumption for our modelled architecture and simulated workloads. *NoCons* represents a (hypothetical) case of running the workloads under no power and thermal constraints. We use three policies, namely, *RoundRobin*, *Alternation*, and *MostFrequentlyUsed* as our baselines due to absence of prior works performing power budgeting for 3D DRAM. *3D-TemPo* dynamically adapts to the workload behavior based on the reward computation and the adjacency-awareness. The channel rewards obtained dynamically represent different workload phases and guide our policy to efficient power allocation. The adjacency-awareness minimizes the thermal impact on 3D DRAM. We report the average execution time improvement of *3D-TemPo* over different baseline policies. Using *3D-TemPo*, we observe an execution time reduction of 23% over *RoundRobin*, 24% over *Alternation*, 25% over MostFrequentlyUsed policies, and 13% over our proposed *Reward-based* policy for a 50W power budget. We observe an average performance improvement of 15% for a 25W budget.

Table 1: Simulation Workload

Suite	Selected Benchmarks	Name	Type
SPLASH-2	lu.cont(×32), water.nsq(×16), radix(×16)	WK-1	mixed
	raytrace(×32), barnes(×16), cholesky(×16)	WK-2	compute
	ocean.cont(×64)	WK-3	memory
SPEC CPU2017	nab(×32), exchange(×16), x264(×16)	WK-4	compute
	lbm(×32), exchange(×16), nab(×16)	WK-5	mixed
	lbm(×64)	WK-6	memory

4.3 Analysis of Policy Behavior

Observation 1: The order of channel activation/deactivation is important. Workloads comprising *multi-threaded* compute-heavy applications (eg. WK-2), with each thread running on a separate core, exhibit sub-optimal performance when all threads are treated uniformly. The channel temperatures remain far below T_{crit}, not requiring DTM. Policies such as *3D-TemPo* and *MostFrequentlyUsed*, prioritising high yielding threads, give best performance.

Observation 2: The order of activation of vertically adjacent channels is important. Workloads comprising *memory-intensive* applications (single or multi-threaded) suffer from severe DTM induced penalty (e.g., WK-3 and WK-6). Selecting a channel amongst the potentially heated vertically adjacent channels in the order of *high reward* ensures maximum progress and eliminates vertical thermal hotspots accumulation. Thus, the adjacency-aware *3D-TemPo* policy gives best performance for such workloads.

Observation 3: The scheduling of applications on processing cores is important. In workloads comprising *mixed* applications, core scheduling decides application-to-channel mapping. A schedule that maps the compute-heavy applications on bottom channels and memory-intensive ones on top channels exhibits lower temperatures and vice-versa. An interleaved scheduling of compute and memory tasks on channels results in moderate heating. We observe that different schedules benefit differently from the same policy.

4.4 Transient Temperature Behavior

Figure 7 shows the transient temperature behavior of a mixed workload (WK-5) for different policies. The workload undergoes heating/cooling cycles due to intense memory activity of *lbm* application. We use an interleaved schedule for the workload. Based on the order of channel activation in policies, the duration between two successive stalls differs. The *NoCons* case undergoes temperatures as high as 140°C. The power budgeting policies are able to prevent thermal violations; however, number of thermal stalls and cooldown time varies. Compared to *RewardBased* policy which prioritizes channel rewards even in DRAM's *critical* region, *3D-TemPo* significantly reduces the thermal stalls by eliminating the vertical thermal hotspots in the *critical* region. For the same reason, *Alternation* outperforms *RewardBased* policy for this workload. The *MostFrequentlyUsed* policy performs similar to *3D-TemPo* due to interleaved schedule, which reduces workload's thermal impact and the penalty of enabling channels based on high memory activity. *RoundRobin* ensures fairness by balancing workload's thermal impact amongst channels, outperforming *RewardBased* policy.

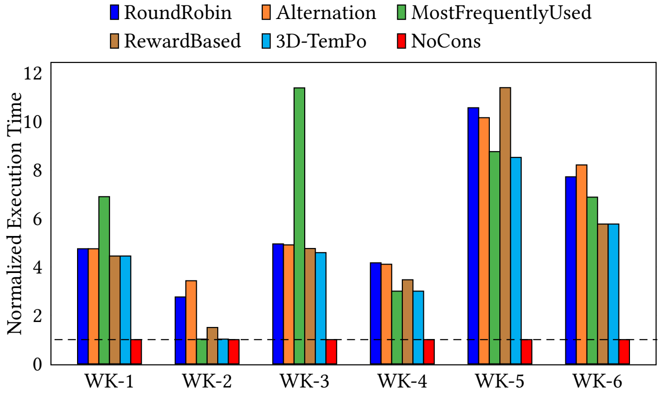


Figure 5: Workload execution time with different policies for P_b=50W and T_{crit}=80°C.

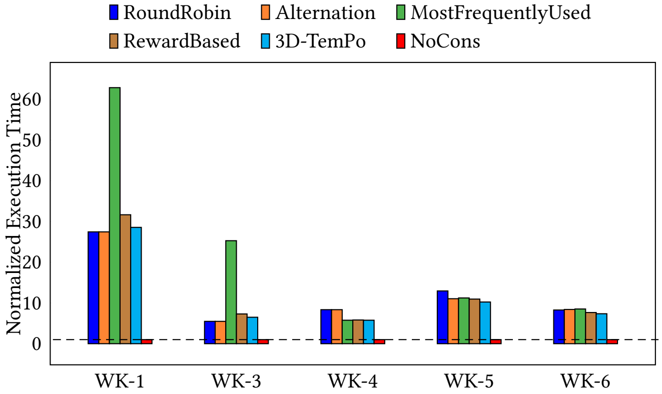


Figure 6: Workload execution time with different policies for P_b=25W and T_{crit}=80°C.

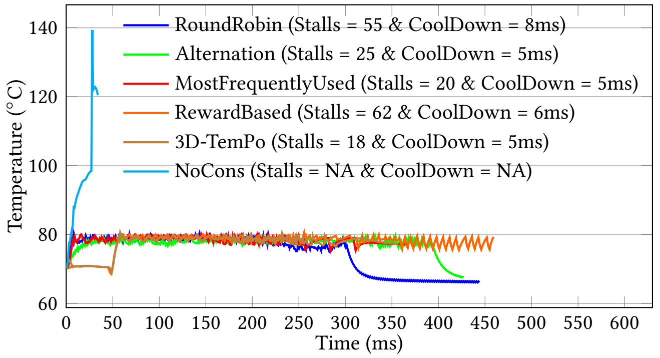


Figure 7: Transient temperature with different power budgeting policies for WK-5.

4.5 Implementation Details

We implement our proposed *3D-TemPo* policy as a software mechanism that periodically sends the rank power states to the memory controller. To measure the channel temperatures, we assume placement of two thermal sensors at each DRAM die [1]. The temperature-dependent leakage power of a channel is looked up in a table that stores P_{leak} at 10°C temperature range obtained using CACTI-3DD. Our workloads exhibit temperatures in [60°C - 80°C] requiring only two table entries and hence negligible storage and lookup time. Computing P_{dyn}, that relies on the DRAM accesses per channel, requires one multiply operation. We estimate the time overhead of *3D-TemPo* by running it on a simulated core, observing a maximum delay of 12μs, negligible compared to the epoch time.

5 CONCLUSION

3D DRAMs are often limited by their power budgets and thermal constraints, resulting in under-utilization of high memory bandwidth. Simplistic power budgeting policies, unaware of physical location of channels and favouring a single metric, do not result in the best performance. We present a heuristic for determining a channel's reward that efficiently captures the system's progress on activating the channel. Further, we leverage the heterogeneity in stacked architecture to minimize associated overheads of power budgeting and thermal management. Our results show an average execution time reduction of 24% over the baseline policies. In the future, we plan to investigate prediction-based and application-aware budgeting policies for 3D DRAMs.

REFERENCES

[1] JEDEC Standard High Bandwidth Memory DRAM (HBM3), JESD238, 2022.
[2] Micron Hybrid Memory Cube – HMC Gen2, 2018.
[3] J. Byrne, "Powerful Hardware and a Strong Software Ecosystem Help Layerscape Excel at AI," https://www.nxp.com, 2018.
[4] H. Wang et al., "New power budgeting and thermal management scheme for multi-core systems in dark silicon," in *SOCC*, 2016.
[5] G. Kornaros, and D. Pnevmatikatos, "Dynamic Power and Thermal Management of NoC-Based Heterogeneous MPSoCs," in *TRETS*, 2014.
[6] O. Sahin, and A. K. Coskun, "On the Impacts of Greedy Thermal Management in Mobile Devices," in *IEEE Embedded Systems Letters*, 2015.
[7] G. Bhat et al., "Algorithmic Optimization of Thermal and Power Management for Heterogeneous Mobile Platforms," in *TVLSI*, 2018.
[8] H. Wang et al., "GDP: A Greedy Based Dynamic Power Budgeting Method for Multi/Many-Core Systems in Dark Silicon," in *TC*, 2019.
[9] A. K. Coskun, J. L. Ayala, D. Atienza, T. S. Rosing, and Y. Leblebici, "Dynamic thermal management in 3D multicore architectures," in *DATE*, 2009.
[10] G. Singla, G. Kaur, A. K. Unver, and U. Y. Ogras, "Predictive dynamic thermal and power management for heterogeneous mobile platforms," in *DATE*, 2015.
[11] J. Meng et al., "Optimizing energy efficiency of 3-D multicore systems with stacked DRAM under power and thermal constraints," in *DAC*, 2012.
[12] C. Zhu, Z. Gu, L. Shang, R. P. Dick, and R. Joseph, "Three-Dimensional Chip-Multiprocessor Run-Time Thermal Management," in *TCAD*, 2008.
[13] Y. Lu, D. Wu, B. He, X. Tang, J. Xu, and M. Guo, "Rank-Aware Dynamic Migrations and Adaptive Demotions for DRAM Power Management" *TC*, 2016.
[14] L. Siddhu et al., "CoMeT: An Integrated Interval Thermal Simulation Toolchain for 2D, 2.5D, and 3D Processor-Memory Systems," in *TACO*, 2022.
[15] S. Kim, W. Kwak, C. Kim, D. Baek, and J. Huh, "Charge-Aware DRAM Refresh Reduction with Value Transformation," in *HPCA*, 2020.
[16] Y. -C. Kwon et al., "25.4 A 20nm 6GB Function-In-Memory DRAM, Based on HBM2 with a 1.2TFLOPS Programmable Computing Unit Using Bank-Level Parallelism, for Machine Learning Applications," in *ISSCC*, 2021.
[17] J. Ahn, S. Yoo, and K. Choi, "Low-Power Hybrid Memory Cubes With Link Power Management and Two-Level Prefetching," in *TVLSI*, 2016.
[18] L. Siddhu, R. Kedia, and P. R. Panda, "Leakage-Aware Dynamic Thermal Management of 3D Memories," in *TODAES*, 2020.
[19] A. Prakash et al., "Improving mobile gaming performance through cooperative CPU-GPU thermal management," in *DAC*, 2016.
[20] A. Pathania, Qing Jiao, A. Prakash, and T. Mitra, "Integrated CPU-GPU power management for 3D mobile games," in *DAC*, 2014.
[21] A. Deshwal et al., "MOOS: A Multi-Objective Design Space Exploration and Optimization Framework for NoC Enabled Manycore Systems," in *TECS*, 2019.