3D-TemPo: Optimizing <u>3D</u> DRAM Performance Under <u>Tem</u>perature and <u>Po</u>wer Constraints

***Abstract*—3D DRAM provides a significant performance boost resulting from substantial 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 their 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, a coordinated strategy for power budgeting and thermal management is needed. We propose an adjacency-aware dynamic power budgeting technique, *3D-TemPo*, which dynamically performs a reward-based power allocation to memory ranks to maximize 3D DRAM performance under power and thermal constraints, and is sensitive to strong thermal correlations between vertically adjacent ranks. We evaluate *3D-TemPo* using *SPEC CPU2017* and *PARSEC 2.1* benchmark suites and observe average execution time improvements of 43% to 10x compared to baseline strategies.**

I. Introduction

Novel memory technologies such as 3D-stacked DRAMs [1], [2], offering substantial memory bandwidth (as high as 386 GBps), have been commercialised in an attempt to break the *memory wall*. High Bandwidth Memory (HBM) [1] is used in modern GPUs and AI processors [3] to meet the bandwidth requirements of memory-intensive workloads involving huge data movement between processing cores and memory. 3D stacking enables hundreds of banks/ranks per memory device; however, the off-chip memory bandwidth is often limited due to its power budget [14]–[16]. Vertical integration results in high power density and poor heat dissipation. Systems running under a power budget and thermal constraint employ dynamic power budgeting and dynamic thermal management (DTM) policies to ensure reliability. Such policies that enable a subset of ranks and affect the performance of 3D DRAM are inter-dependent and cannot be performed in isolation. The static/leakage power of memory shows an exponential rise with increasing temperature, motivating the need for thermal-aware power budgeting. 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 (Fig. 1) with varying 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. 3D DRAM exhibits a strong thermal correlation between vertically adjacent DRAM ranks/channels compared to those within the same die. This thermal gradient motivates us to propose an adjacency-aware dynamic power budgeting strategy. We make the 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 and thermal shutdown (due to high dynamic power dissipation), thereby stalling the cores, and (3) enabling vertically adjacent high memory activity channels could impede performance, as vertically aligned thermal hotspots drastically reduce 3D DRAM's cooling efficiency.

In this work, we make the following specific contributions:

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

II. Related Work

With increasing power density in novel memory technologies, power budgeting and thermal analysis of 3D architectures becomes important [18]–[20]. Prior works have proposed solutions to efficient power budgeting and thermal management primarily for homogeneous and heterogeneous multicore systems using task migration and DVFS [4], workload's power profile [5], QoS-aware frequency throttling [6], temperature prediction [7], [9], and avoiding the clustering of active cores [21]. For power and thermal management in 3D multicore architecture (multiple layers of cores stacked together), Coskun et al. [8] proposed job scheduling and DVFS, Zhu et al. [11] leveraged the heterogeneous thermal characteristics of cores in different layers, and Meng et al. [10] used DVFS on cores to optimize energy efficiency, assuming uniform traffic across all DRAM banks. FastCool [17] is a channel turn ON/OFF strategy that migrates data to a backup 2D DRAM upon thermal emergencies in 3D DRAM, without considering memory power budget. We target the power budgeting problem under thermal constraint for 3D DRAM, assuming different subsets of cores mapped to different memory channels, which generates non-uniform traffic across different 3D DRAM banks depending on the workload's run-time behavior.

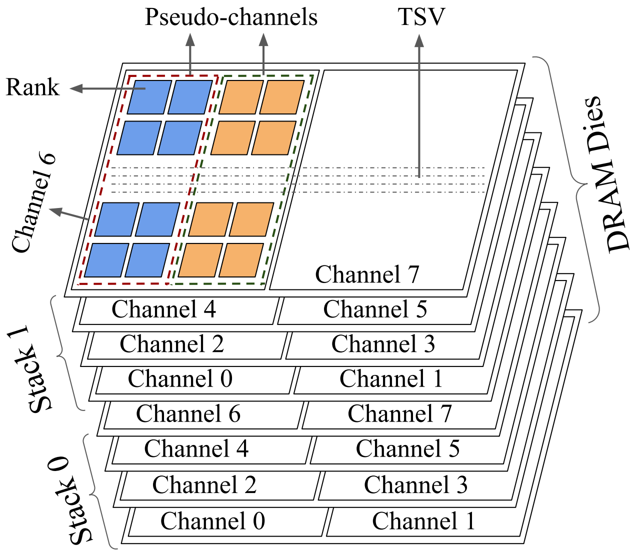

Fig. 1: High Bandwidth Memory (HBM2E) structure

III. 3D DRAM Power Budgeting and Thermal Management

A. Memory Architecture

A 3D DRAM consists of multiple stacks, with each stack comprising several DRAM dies stacked vertically, connected using fast through-silicon via (TSV). A portion of the DRAM die from each stack is connected to the same independent physical channel, each channel consisting of two pseudo-channels with multiple ranks, which in turn contain banks that are organised into rows and columns, similar to a conventional 2D DRAM. Figure 1 shows the structure of the popular HBM2E with 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 (0-7) services a total of 16 banks from two DRAM dies located in Stacks 0 and 1. A subset of CPU cores is mapped to each memory channel, restricting the memory traffic generated by each core 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 and leakage). The *standby* state, which does not permit memory accesses, consumes a fraction of the leakage power, required only to retain the data. In the *active* state, the memory consumes higher leakage power, and is ready for accesses. The power state transition of ranks requires a minimal overhead (typically of the order of a few ns to a few μs) [12]. Given a power budget, our dynamic power budgeting policy allocates power at the level of a memory channel. 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.

B. Problem Definition

The 3D DRAM thermal-constrained memory power budgeting (MPB) problem is formulated as follows. Given: (1) a multi-core processor with p processing cores, (2) a 3D DRAM memory with c channels, each channel mapped to p/c cores, and (3) a memory power budget P_b, and temperature constraint T_{crit}, we aim to minimize the workload's total execution time (i.e., maximize progress), subject to:

$$T_{\text{i}} \leq T_{\text{crit}}, i \in \{0, 1, ..., (c-1)\}, and \\ \sum_{i \in \{0,1,...,(c-1)\}} P_i \leq P_b \tag{1}$$

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

The system execution is divided into intervals, and the MPB problem is defined in terms of the power consumption and progress (memory access throughput) in the last interval the channel was active, the solution being used to define the memory configuration for the next interval. The MPB problem can be shown to be NP-Hard using a reduction from the well-known NP-Complete problem, *Knapsack*. Consider an instance of the Knapsack problem with n objects, with object i characterized by weight w_i and value v_i, and a maximum weight budget W. The problem is to choose a subset $S \subseteq \{1, ..., n\}$ such that $\sum_{i \in S} w_i \leq W$ and $\sum_{i \in S} v_i$ is maximized. We construct an MPB instance with channel i corresponding to object i, *progress* (measured as IPC of the cores mapped to this channel in the last interval when the channel was active) $q_i = v_i$, power consumption P_i of channel i in the last active interval $= w_i$, power budget $P_b = W$, $T_{\text{crit}} = \infty$, and the subset of channels activated for the next interval corresponding to the selected subset S. Since a large T_{crit} value makes the temperature correlation effects across channels irrelevant, an efficient solution to the MPB problem directly solves the Knapsack problem due to the one-to-one correspondences, making MPB an NP-Hard problem.

C. Preliminary Power Budgeting Policies

Assuming a 3D DRAM with 8 channels and defining an interval (or epoch) as the duration between two consecutive invocations of a policy, we discuss the working of simplistic dynamic power budgeting policies below:

1) *Round Robin Policy* activates Channels 0 to 3 in the even intervals, and Channels 4 to 7 in the odd intervals of workload execution, ensuring fairness amongst all the memory channels. However, it treats all the cores uniformly and ignores the strong thermal correlation between vertically adjacent channels.
2) *Alternation Policy* allocates power to channels in alternate DRAM dies in both even and odd intervals, eliminating vertically adjacent thermal hotspots. While this helps in faster DRAM cooling, and ensures fairness, the memory activity rate in channels is ignored. The policy is an adaptation of [21], avoiding clustering of active channels in the vertical direction.
3) *Most Frequently Used (MFU) Policy* assigns power to the most frequently used channels where a channel's frequency is computed based on its last active interval. It maintains an *MFU* queue of channels, selecting channels

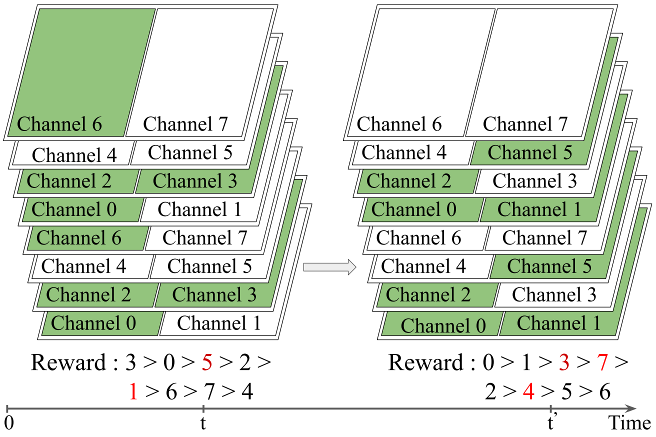

Fig. 2: Working of *Adjacency-aware 3D-TemPo* policy

from the head of the queue, leading to high memory throughput. However, it leads to a sharp rise in frequently used channels' temperatures due to prolonged activation. It also results in the accumulation of vertical hotspots, and starvation of *compute-heavy* cores, as the channels mapped to them are least preferred for activation.

IV. The 3D-TempPo Approach

In this section, we present our proposed *3D-TemPo* policy that jointly considers (1) a memory channel's activity, (2) the *core progress* obtained through activating that channel, and (3) its physical location while allocating the power budget. We discuss the *reward* computation strategy, the main ideas of *3D-TemPo*, the DRAM low-power states based DTM, and the overall flow in the following sections.

A. Reward Computation

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 the current interval as the ratio of its *profit* to its *weight* in the interval when it was *last* activated, 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 V-E), and (3) $P_{ref}(j)$ is taken to be a constant.

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

B. Incorporating Adjacency-awareness

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 in their last active interval. As 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 *3D-TemPo* uses the *channel reward* (Eq. 2) considering both IPC of the cores and the DRAM's thermal condition. The channel rewards are sorted in the non-increasing order, activating the channels in that order. To prevent vertically aligned thermal hotspots and improve the cooling efficiency we leverage adjacency-awareness in *3D-TemPo* for the *critical* region of 3D DRAM. The adjacency-awareness additionally exploits the observed thermal correlation between vertically adjacent channels. 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 already activated channels. This prevents the trapping of heat between vertically adjacent channels, minimizing DTM penalty. We skip a high reward channel only when its vertically adjacent neighbours n are in the *critical region* ($T_{hot} \leq T_n \leq T_{crit}$). Figure 2 illustrates the strategy, with channel rewards listed at the bottom. At time t, Channel 5 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 $T_{ch0} < T_{hot}$. 3D-TemPo also prevents starvation of low reward channels by activating them at coarser time intervals.

Unlike *Round Robin*, *3D-TemPo* prioritizes *compute-heavy* cores, ensuring maximum system progress with minimum memory power dissipation. Unlike *MFU*, it selects the channels based on their *rewards* and not merely on the instantaneous memory throughput. We incorporate a channel's physical location in 3D stack by considering the temperature-dependent leakage power. Top channels undergo a slow temperature rise, consuming less leakage power than the bottom channels even when subjected to similar memory access rates. *3D-TemPo* prioritizes top channels over bottom ones when similar *progress* is expected in the corresponding cores, reducing thermal stalls.

C. DRAM Low-power Based DTM

As discussed earlier, modern DRAMs allow rank-level control on the power states to help manage power dissipation. We employ the DRAM power states to also keep the memory channel temperatures under the thermal limit. Our policy performs the DTM at the channel level, whereby, a channel is sent to low-power *standby* state (causing a thermal stall) if any of its ranks exceeds T_{crit}. Similarly, a channel is reverted to *read/write* state upon cooling down of all its ranks. During thermal emergencies, the power budgeting policy does not consider the channel for power allocation regardless of the associated reward or priority. The DTM policy sends the temperature of memory channels to the budgeting policy at the start of each interval.

D. The Overall Flow

The *3D-TemPo* policy (Algorithm 1) uses low-power states based DTM for ensuring safe thermal limits and performs

```
Algorithm 1: Adjacency-aware 3D-TemPo Policy
   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 ($P_b$, $Activated$)
7  else // Hot or critical region.
8      for each channel ch in {0, 1, ... , (c-1)} do
           // Compute reward using Eq. 2
9          $Reward[ch] \leftarrow$ Compute_Channel_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_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 ($ch$, $Activated$)
                   // Update Power Budget.
17                 $P_b$ -= $P_{ch}$

   // Place recovered channels to
      read/write state
18 $RankState \leftarrow$ Set_Active_On_Recovery ($T$,$T_{rec}$)
19 return $RankState$, $Activated$
```

reward-based and adjacency-aware power allocation to memory channels, leveraging the thermal gradient in 3D DRAM. 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 T_{crit} (temperature constraint), the DTM policy is invoked and appropriate rank states (*RankState*) are computed (Lines 2-4). Potentially available channels that are not yet set to *standby* state, 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 $\geq T_{hot}$), activated upon *non-critical* vertical neighbours, and skipped otherwise, updating the power budget (Lines 11-17). Upon cooling down of all channel ranks below T_{rec}, the channel is reset to *read/write* 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.

TABLE I: Simulation Workloads

Suite	Selected Benchmarks	Name	Type
SPEC 2017	lbm(×32)	WK-1	memory
	mcf(×32)	WK-2	memory
	nab(×16), x264(×8), exchange(×8)	WK-3	compute
(single-threaded)	exchange(×8), nab(×8), lbm(×16)	WK-4	mixed
	lbm(×24), gcc(×8)	WK-5	memory
	mcf(×8), lbm(×24)	WK-6	memory
	nab(×8), mcf(×8), gcc(×8), lbm(×8)	WK-7	mixed
	x264(×16), exchange(×16)	WK-8	compute
	lbm(×16), mcf(×16)	WK-9	memory
	gcc(×8), x264(×8), lbm(×8), exchange(×8)	WK-10	mixed
PARSEC 2.1	blackscholes(×32)	WK-11	compute
	bodytrack(×32)	WK-12	mixed
	fluidanimate(×32)	WK-13	mixed
(multi-threaded)	streamcluster(×32)	WK-14	memory
	swaptions(×32)	WK-15	compute

V. Experimental Evaluation

A. Simulation Environment

We use an integrated performance-thermal simulator, CoMeT [13], consisting of Sniper 7.2 multicore simulator and Hotspot 6.0, for 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 [17]. Computed temperatures are sent back to Sniper, where the dynamic power budgeting and thermal management decisions are implemented.

We model a multi-core processor (32 cores, 3.6GHz, 22 nm, out-of-order, caches: 32KB private L1, 256 KB private L2, 32 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 32 applications/threads, one on each core. The *standby* state consumes 17% static power and the transition overhead to *active* state is ∼6μs [12], negligible compared to the epoch time. We empirically determine the temperature thresholds by using different values and observing the performance gains: T_{cool}=74°C, T_{hot}=78°C, T_{rec}=77°C, and T_{crit}=80°C (similar to [17]).

We use a diverse set of workloads from *SPEC CPU2017* and *PARSEC 2.1* benchmark suites to validate the efficacy of our proposal. Table I presents the workload details and their characteristics. We attempted to cover a wide range of benchmark mixes with varying memory access characteristics. We

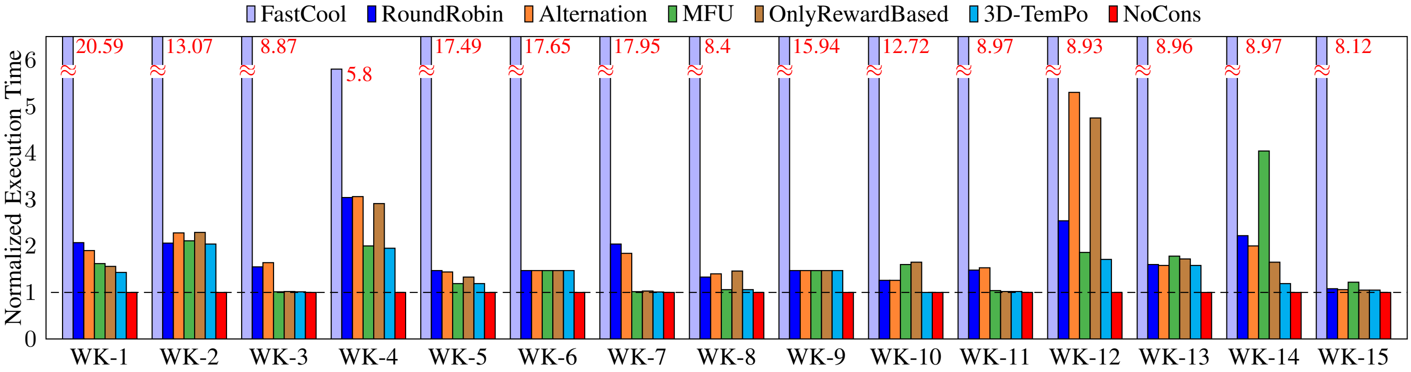

Fig. 3: Execution time of workloads with different power budgeting policies for P_b=64W and T_{crit}=80°C, normalized to no power and thermal constraints (NoCons).

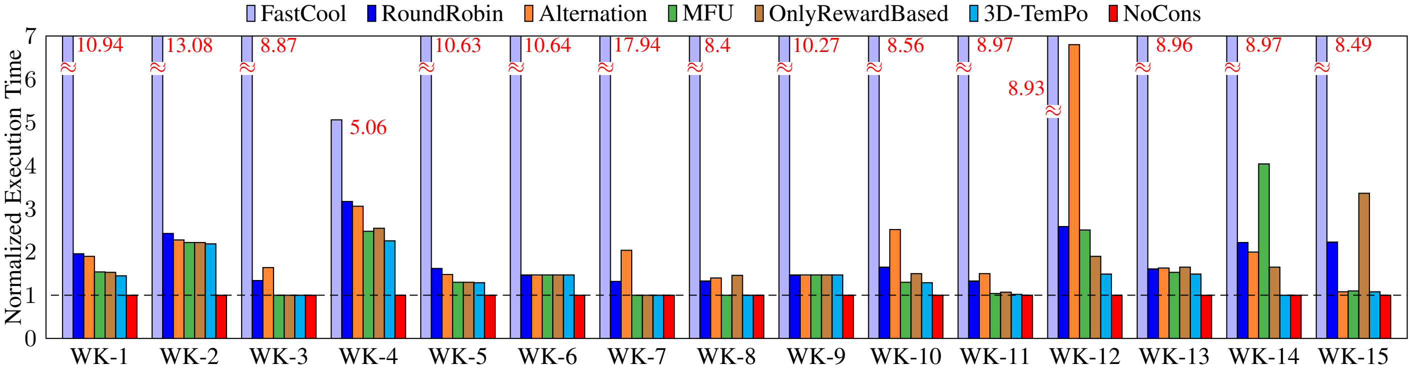

Fig. 4: Execution time of workloads with different power budgeting policies for P_b=96W and T_{crit}=80°C, normalized to no power and thermal constraints (NoCons).

simulate the compiled source code for PARSEC 2.1 workloads with input size *simlarge* and pre-generated traces (Pinballs) for 100M instructions for SPEC CPU2017.

B. Performance Improvement

Figures 3 and 4 show the execution time (normalized to *NoCons*) of different policies for two different power budgets, 64W and 96W, accounting for 50% and 75% 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 five baseline policies: (1) *FastCool* [17] augmented with a power budgeting logic, allocating the power budget amongst channels turned ON by *FastCool*, (2) *RoundRobin*, (3) *Alternation*, a memory power budgeting variant of [21], (4) *MFU*, and (5) only *RewardBased* component of *3D-Tempo*. We report the average execution time improvement of *3D-TemPo* over different baseline policies. *FastCool* performs poorly under limited power budget due to the associated data migration costs. We consistently observe better or similar performance with *3D-TemPo* for all workloads and all baselines and achieve an average execution time reduction of 43% over *RoundRobin*, 60% over *Alternation*, 28% over *MFU*, 41% over only *OnlyRewardBased* component, and 10.81x over *FastCool* for a 64W power budget. We observe an average performance improvement of 8.38x over *FastCool*, and 45% over other baselines for a 96W budget that allows more channel activations. *3D-TemPo* dynamically adapts to the workload phases by computing rewards and leveraging adjacency-awareness, minimizing the overall thermal impact.

C. Analysis of Policy Behavior

Observation 1: The order of channel activation/deactivation is important. Workloads comprising *multi-threaded* compute-heavy applications (e.g., WK-11 and WK-15), 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 *MFU*, prioritising high yielding threads, give best performance. The *single-threaded* compute-heavy workloads (eg., WK-3 and WK-8) also benefit equally from *3D-TemPo* and *MFU*.

Observation 2: The order of activation of vertically adjacent channels is important. Workloads comprising *memory-intensive* applications (single or multi-threaded) with high memory access rates suffer from severe DTM induced penalty (e.g., WK-1, WK-2, WK-5, and WK-14). Selecting a channel amongst the potentially heated vertically adjacent channels in the order of *high reward* ensures maximum progress and eliminates vertical thermal hotspots. Thus, the adjacency-aware *3D-TemPo* performs best for such workloads.

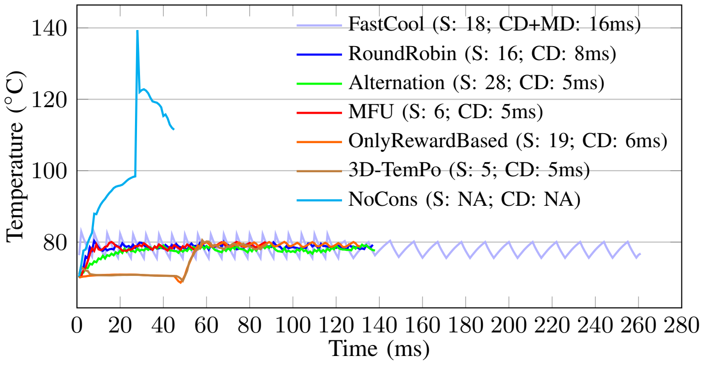

Fig. 5: Transient temperature for *WK-4* with different power budgeting policies. P_b=64W, *S* denotes Stalls, *CD* denotes CoolDown time, and *MD* denotes migration delay in *FastCool*.

Observation 3: The scheduling of applications on processing cores is important. In workloads comprising *mixed* applications (e.g., WK-4, WK-7, WK-10, WK-12, and WK-13) or different *memory-intensive* applications with diverse memory-access rates (WK-6 and WK-9), core scheduling decides application-to-channel mapping. A schedule that maps the compute tasks on bottom channels and memory-intensive ones on top channels exhibits lower temperatures and vice-versa. Interleaving compute and memory tasks on channels results in moderate heating. We observe that different schedules benefit differently from the same policy.

D. Transient Temperature Behavior

Figure 5 shows the transient temperature behavior of a *mixed* workload (WK-4) 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. The duration between successive stalls varies with the order of channel activation in the policies. 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 times are widely varying. Compared to *OnlyRewardBased* which prioritizes channel rewards even in the DRAM's *critical* region, *3D-TemPo* significantly reduces thermal stalls by avoiding the vertical thermal hotspots in the *critical* region. For the same reason, *OnlyRewardBased* only marginally outperforms *Alternation*, and *RoundRobin* policies. *MFU* and *3D-TemPo* have similar performance due to the interleaved schedule, which reduces the workload's thermal impact and the penalty of enabling high frequency channels. *FastCool* undergoes frequent data migration to 2D memory due to power budgeting and DTM, and incurs maximum overheads.

E. Implementation Details

We implement *3D-TemPo* policy as a software mechanism that periodically sends the rank power states to the memory controller. To measure channel temperatures, we assume the 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-wide temperature ranges 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}, which uses the DRAM access count 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, which is negligible compared to the epoch time.

VI. 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 adjacency-awareness to minimize associated overheads of power budgeting and DTM. Our results show an average execution time reduction of 43% to 10x 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 *Embedded Systems Letters*, 2015.

[7] G. Bhat et al., "Algorithmic Optimization of Thermal and Power Management for Heterogeneous Mobile Platforms," in *TVLSI*, 2018.

[8] A. K. Coskun et al., "Dynamic thermal management in 3D multicore architectures," in *DATE*, 2009.

[9] G. Singla et al., "Predictive dynamic thermal and power management for heterogeneous mobile platforms," in *DATE*, 2015.

[10] J. Meng et al., "Optimizing energy efficiency of 3-D multicore systems with stacked DRAM under power and thermal constraints," in *DAC*, 2012.

[11] C. Zhu, Z. Gu, L. Shang, R. P. Dick, and R. Joseph, "Three-Dimensional Chip-Multiprocessor Run-Time Thermal Management," in *TCAD*, 2008.

[12] Y. Lu et al., "Rank-Aware Dynamic Migrations and Adaptive Demotions for DRAM Power Management" *TC*, 2016.

[13] L. Siddhu et al., "CoMeT: An Integrated Interval Thermal Simulation Toolchain for 2D, 2.5D, and 3D Processor-Memory Systems," in *TACO*, 2022.

[14] S. Kim, W. Kwak, C. Kim, D. Baek, and J. Huh, "Charge-Aware DRAM Refresh Reduction with Value Transformation," in *HPCA*, 2020.

[15] 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.

[16] J. Ahn, S. Yoo, and K. Choi, "Low-Power Hybrid Memory Cubes With Link Power Management and Two-Level Prefetching," in *TVLSI*, 2016.

[17] L. Siddhu, R. Kedia, and P. R. Panda, "Leakage-Aware Dynamic Thermal Management of 3D Memories," in *TODAES*, 2020.

[18] A. Prakash et al., "Improving mobile gaming performance through cooperative CPU-GPU thermal management," in *DAC*, 2016.

[19] A. Pathania, Qing Jiao, A. Prakash, and T. Mitra, "Integrated CPU-GPU power management for 3D mobile games," in *DAC*, 2014.

[20] A. Deshwal et al., "MOOS: A Multi-Objective Design Space Exploration and Optimization Framework for NoC Enabled Manycore Systems," in *TECS*, 2019.

[21] H. Wang et al., "DBP: Distributed Power Budgeting for Many-Core Systems in Dark Silicon," in *TCAD*, 2022.