Aging-Aware Request Scheduling for Non-Volatile Main Memory

Shihao Song
Drexel University
USA

Onur Mutlu
ETH Zürich
Switzerland

Anup Das
Drexel University
USA

Nagarajan Kandasamy
Drexel University
USA

ABSTRACT
Modern computing systems are embracing non-volatile memory (NVM) to implement high-capacity and low-cost main memory. Pervasive device scaling increases power density and temperature, which further accelerates aging, challenging the reliable operation of NVM-based main memory. We propose HEBE, an architectural technique to mitigate the circuit aging-related problems of NVM-based main memory. HEBE is built on three contributions. First, we propose a new analytical model that can dynamically track the aging in the peripheral circuitry of each memory bank based on the bank’s utilization. Second, we develop an intelligent memory request scheduler that exploits this aging model at run time to de-stress the peripheral circuitry of a memory bank only when its aging exceeds a critical threshold. Third, we introduce an isolation transistor to decouple parts of a peripheral circuit operating at different voltages, allowing the decoupled logic blocks to undergo long-latency de-stress operations independently and off the critical path of memory read and write accesses, improving performance. We evaluate HEBE with workloads from the SPEC CPU2017 Benchmark suite. Our results show that HEBE significantly improves both performance and lifetime of NVM-based main memory.

ACM Reference Format:
Shihao Song, Anup Das, Onur Mutlu, and Nagarajan Kandasamy. 2021. Aging-Aware Request Scheduling for Non-Volatile Main Memory. In 26th Asia and South Pacific Design Automation Conference (ASPDAC ’21), January 18–21, 2021, Tokyo, Japan. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3394885.3431529

1 INTRODUCTION
DRAM has been the technology choice for implementing main memory due to its relatively low latency and low cost. However, DRAM is a fundamental performance and energy bottleneck in almost all computing systems, and it is experiencing significant technology scaling challenges [22, 36, 40, 54, 59–61]. Recently, DRAM-compatible, yet more technology-scalable alternative non-volatile memory (NVM) technologies such as Phase-Change Memory (PCM), are being explored [44, 46–48, 57, 64, 69–71, 81, 84, 85, 94, 100].

Compared to DRAM, NVM requires higher voltages to read and program memory cells. We investigate the internal architecture of the peripheral circuitry of each memory bank and find that such circuitry consists of transistors built using CMOS and FinFET [28]. When operated at high voltage and temperature, over time the transistor’s parameters can strongly drift from their nominal values. This is called aging. In fact, in scaled technology nodes, aging happens even under nominal conditions from the very start of device use. The most important breakdown mechanism is the Bias Temperature Instability (BTI) [43, 92]. Strongly depending on the workload, BTI is highly variable and largely reversible under nominal conditions upon removal of the stress voltage. However, if the peripheral circuitry is used continuously for long durations at elevated operating conditions, the BTI induced parameter drifts in peripheral circuitry cannot be reversed [98], leading to permanent functional degradation and hardware faults.

As process technology scales down to smaller dimensions due to NVM’s CMOS-compatible scaling [96], aging issues are expected to get exacerbated due to the increase in the electric field and power density, which leads to higher chip temperatures and, consequently, the acceleration of BTI. Current methods for improving aging are overly conservative, since they estimate transistor aging in a peripheral circuitry statically, assuming worst-case operating conditions [31]. Based on such worst-case estimates, these methods de-stress each peripheral circuitry periodically at a fixed interval, without tracking its actual aging. Therefore, these methods significantly and unnecessarily constrain performance.

Our goal is to design a dynamic policy to track the aging in the peripheral circuitry of each memory bank based on the operating voltages needed to serve read and write requests from the bank, and dynamically schedule its de-stress operation only when its aging exceeds a critical threshold. Our architectural approach to mitigate aging in NVM, called HEBE, is built on three contributions.

Contribution 1. We develop a new, accurate analytical model to estimate transistor aging in peripheral circuitry of each memory bank. Our model dynamically tracks aging in response to a memory controller’s request scheduling decisions such as serving a read (which requires 2.85V) vs. a write (which requires 3.7V). To use this model at run time, we leverage the associative property of our analytical formulation, a direct reflection of the underlying

---

1NVMs are also used for synaptic storage in neuromorphic computing [4, 5, 8, 53, 82].
2In Greek mythology, Hebe (pronounced hee bee) is the goddess of youth [91].
physical failure mechanism, allowing us to express aging in terms of offline-computed unit aging parameters (described in Section 3). Our memory controller uses these parameters to estimate aging in a peripheral circuitry based on the number of read and write requests that are served via the circuitry.

**Contribution 2.** We develop a new, intelligent memory request scheduler that prioritizes requests to banks whose peripheral circuits are currently active but not serving any memory request, over other requests, including the long-outstanding ones. This straightforward and greedy policy is controlled in two ways. First, the memory controller uses our new aging model to track the aging of a peripheral circuitry, de-stressing the circuitry only when its aging exceeds a critical threshold. Second, the memory controller uses thresholding to avoid starvation of memory requests.

**Contribution 3.** We introduce an isolation transistor in each peripheral circuitry to decouple its logic blocks operating at different supply voltages during read and write accesses (see Fig. 1). The decoupled architecture allows these logic blocks to be de-stressed based on their respective aging levels. Our request scheduler exploits this decoupled architecture to schedule the long-latency de-stress operations off the critical path of accesses, reducing bank occupancy and improving performance.

We evaluate HEBE with workloads from the SPEC CPU2017 Benchmark suite [7]. Our results show that HEBE significantly improves both performance and lifetime of NVM-based main memory.

### 2 BACKGROUND

NVM, like DRAM [22, 42, 49, 78], is organized hierarchically [56, 58, 73, 81, 84, 85, 100]. For example, a 128GB NVM can have 2 channels, 1 rank/channel and 8 banks/rank. A bank can have 64 partitions [81]. Each bit in NVM is represented by the resistance of an NVM cell: low resistance is logic ‘1’ and high resistance is logic ‘0’. An NVM cell is read and programmed by driving current through it using per bank peripheral circuitry (see Fig. 1). Peripheral circuitry consists of sense amplifiers (SA) to read and write drivers (WD) to program. WD consists of the write pulse shaper (PS) logic, which generates the current pulses necessary for SET and RESET operations, and the verify (VF) logic, which verifies the correctness of these operations [25].

In addition to the regular read and write mode of operations, peripheral circuitry can also be in 1) idle mode, where it does not serve any request, and 2) de-stress mode, where it is powered down. Table 1 reports the operating voltages of the three logic blocks in a peripheral circuitry during read, write, idle, and de-stress operations [73]. Voltages higher than the nominal 1.2V supply are generated using the two on-chip charge pumps shown in Figure 1. These high voltages induce aging of the transistors in the peripheral circuitry logic blocks. We focus on BTI failures.

**Figure 2:** Threshold voltage ($V_{th}$) shift due to BTI.

BTI is a failure mechanism in a transistor, where positive charge is trapped in the oxide-semiconductor boundary underneath the gate [26]. BTI manifests as 1) decrease in drain current and transconductance, and 2) increase in off current and threshold voltage $V_{th}$. Figure 2 illustrates the stress and recovery of the threshold voltage of a transistor on application of a high ($V_{read}/V_{write}$) and a low ($V_{de-stress}$) voltage. We observe that both stress and recovery depends on the time of exposure to the corresponding voltage level. This implies that when peripheral circuitry is de-stressed, the BTI aging of its transistors partially recovers from stress. To compute the overhead due to de-stress operations, we assume that the memory controller issues a de-stress command to a memory bank once every $t_{DSI}$, the de-stress interval. Each de-stress operation completes within a time interval $t_{DSC}$, the de-stress cycle time. Hence, the performance overhead (i.e., data throughput loss) due to periodic de-stress is

$$\text{de-stress overhead} = \frac{t_{DSC}}{t_{DSI}}.$$

The overhead due to periodic de-stress (as implemented in conservative approaches such as [31]) is significant in current NVM devices, and it is expected to become even more performance-critical in the future as NVM chip capacity increases [73].

| Operating mode | Operating voltage |
|----------------|-------------------|
| Read           | 1.2V              |
| Write (program)| 3.7V              |
| Idle           | 1.2V              |
| De-stress      | $< V_{th}$        |

Table 1: Operating voltage of the three logic blocks in peripheral circuitry during read, write, idle, and de-stress [73]. The threshold voltage ($V_{th}$) of a CMOS transistor is between 0.7V and 1V at scaled nodes.

Figure 3 shows the shift in threshold voltage of a transistor in a memory bank’s peripheral circuitry when executing a microbenchmark with $t_{DSI}$ set to 10 and 100 requests, respectively. We observe that the shift in the threshold voltage is higher for larger de-stress interval $t_{DSI}$. This is because when we set $t_{DSI}$ to a large value, the transistor is exposed to a stress voltage for a long duration between two consecutive de-stress operations. Therefore, the large parameter drift it encounters in this duration cannot be reversed during the next de-stress operation. The parameter drift

1The microbenchmark we use for this evaluation consists of alternating read and write requests to randomly selected PCM locations.
continues to accumulate over time, resulting in a large shift in the threshold voltage as shown in the figure. Therefore, it is better to set the $t_{DSI}$ to a small value, which allows most of the parameter drifts to be reversed during each de-stress operating, resulting in a lower threshold voltage shift.

![Shift in $V_{th}$ for $t_{DSI} = 10$ and 100.](image)

However, setting a lower $t_{DSI}$ leads to a higher de-stress related performance overhead, which we formulate in Equation 1. HEBE exploits this performance-reliability trade-off using its new intelligent memory request scheduler (see Sec. 5).

### 3 NEW AGING MODEL OF HEBE

In this section, we introduce the new aging model of HEBE. The BTI lifetime [3, 17–20, 80, 83, 85] of a transistor is

\[
\text{MTTFTI} = \frac{A}{\gamma} e^{\frac{E_a}{kT}}
\]

where $A$ and $\gamma$ are material-related constants, $E_a$ is the activation energy, $k$ is the Boltzmann constant, $T$ is the temperature, and $V_i$ is the overdrive gate voltage of the transistor.\(^4\) BTI failures can also be modeled using the Weibull distribution with a scale parameter $\alpha$ and a shape parameter $\beta$. The reliability, defined as the probability of correct operation of the transistor, at time $t$ is given by [6, 21, 29, 86]

\[
R(t) = e^{-\left(\frac{t}{\alpha}\right)^\beta}
\]

with the corresponding MTTF computed as

\[
\text{MTTF} = \int_0^\infty R(t) dt = \alpha(V_i) \Gamma\left(1 + \frac{1}{\beta}\right)
\]

where $\Gamma$ is the Gamma function. Using the expressions for MTTF from Equations 2 and 3, and rearranging, we obtain the expression for the scale parameter $\alpha$ as

\[
\alpha(V_i) = \frac{A}{\gamma} e^{\frac{E_a}{kT}} \Gamma\left(1 + \frac{1}{\beta}\right)
\]

Figure 4 shows the operating voltage of PS, VR, and SA logic blocks in the peripheral circuitry of a memory bank when serving read and write requests. (Overdrive voltage = operating voltage - $V_{th}$).

At the end of the first interval (i.e., after servicing the first read request), the reliability of the PS logic is

\[
R(t_1) = e^{-\left(\frac{t_1}{\alpha(V_i)}\right)^\beta}
\]

Using the term $\theta$ to represent reliability degradation during this interval [$t_0, t_1$], the reliability at the beginning of the second interval (i.e., right after the start of the first idle period) is

\[
R(t_2) = e^{-\left(\frac{t_2 + \theta}{\alpha(V_i)}\right)^\beta}
\]

Due to the continuity of the reliability function, we can equate Equations 7 & 8 to compute $\theta$ as

\[
\theta = \left(\frac{\alpha(V_i)}{\alpha(V_i)}\right)^\beta - 1 t_1
\]

Substituting Eq. 9 in Eq. 8, reliability at time $t_2$ is

\[
R(t_2) = e^{-\left(\frac{t_2}{\alpha(V_i)}\right)^\beta}
\]

We can extend this equation to compute the reliability of the PS logic at the end of execution (i.e., after servicing the last write request from the bank in Fig. 4) as

\[
R(t_n) = e^{-\left(\frac{t_n}{\alpha(V_i)}\right)^\beta}
\]

The aging $\mathcal{A}_{PS}$ of the PS logic is

\[
\mathcal{A}_{PS} = \sum_{i=1}^n \frac{\Delta t_i}{\alpha(V_i)} \text{ such that } R(t_n) = e^{-\mathcal{A}_{PS}^\beta}
\]

where the scaling factor $\alpha(V_i)$ can be calculated using Eq. 5.

We observe that Eq. 12 follows the associative property, a direct reflection of the underlying BTI failure mechanism. In other words, the aging accrued in each bank’s peripheral circuit is independent of the order in which the reads and writes are scheduled to the bank. Eq. 12 can be rewritten using memory timing parameters as

\[
\mathcal{A}_{PS} = n_r \cdot U_r + n_w \cdot U_w + n_{r+w} \cdot U_{r+w}
\]

where $t_{RC}$ is the row cycle time, $n_r$ and $n_w$ are the number of read and write requests, respectively, and $n_{r+w}$ is the number of memory clock cycles for which the PS logic is idle. $U_r$ and $U_w$ represent respectively, the aging accrued in peripheral circuitry when serving a read and a write request, and $U_{r+w}$ represents the aging accrued per clock cycle when the peripheral circuit is idle. $U_r$, $U_w$, and $U_{r+w}$ are called unit aging parameters, which the memory controller
uses to track the aging of the PS logic in peripheral circuitry by simply recording 1) the number of read and write requests that are served from the bank, and 2) the number of idle clock cycles during workload execution. We note that these factors (read/write requests and the idle periods) cannot be known with certainty at design-time. Therefore, design-time aging estimates are not accurate.

The aging of VR and SA logic blocks (represented as $A_{VR}$ and $A_{SA}$, respectively) can be computed in a similar way using Eq. 13. To obtain the overall aging, we combine these individual aging values considering the peripheral circuitry to be a series failure system, where the first instance of any logic block failing causes the entire peripheral circuit to fail. Therefore, the overall aging is

$$A = \max\{A_{PS}, A_{VR}, A_{SA}\}. \quad (14)$$

## 4 DECOUPLED PERIPHERAL CIRCUIT ARCHITECTURE OF HEBE

In the baseline system, when peripheral circuitry is de-stressed, its three logic blocks (PS, VR, and SA) are de-stressed simultaneously. Once de-stressed, these logic blocks take several memory cycles ($t_{DSC}$) before they can be used to serve memory requests again. In recent designs, $t_{DSC} = 10$ cycles [73]. Therefore, frequent de-stress operations can lead to high performance overhead (Eq. 1). To reduce this overhead, we analyze the average aging of the PS, VR, and SA blocks in a memory bank’s peripheral circuitry at the time when they are de-stressed during workload execution. Figure 5 plots these results for the workloads described in Section 6 with $t_{DSI}$ set to 100.

We make the following two key observations.

![Figure 5: Average aging in arbitrary units (a.u.) of the PS, VR, and SA logic blocks in peripheral circuitry, at the time when they are de-stressed for our evaluated workloads.](image)

Therefore, when HEBE de-stresses the SA because SA’s aging exceeds the critical aging threshold, the VR logic also gets de-stressed, preventing the write driver from serving write requests. To address this, we exploit the decoupled program and verify-based write operations in PCM [25]. If a write request needs to be scheduled concurrently with the de-stress operation of SA, HEBE schedules only the program step of the write operation (which utilizes the PS block) concurrently with the de-stress operation, while the verify step is scheduled after the de-stress operation completes.

| Charge Pump Control | Peripheral Circuit Action |
|---------------------|---------------------------|
| Read    | Write | PS | VR  | SA  |
|----------------------------------------------------------------------|
| **Active** | **Active** | **Active** | **Active** | **Active** |

| Proposed Decoupled Control |
|----------------------------|
| Active | Active | Active | Active | Active |
| Discharged | Discharged | De-stress | De-stress | De-stress |

| Discharged | Active | Discharged | Active | Active | Active |
| Active | Discharged | Active | De-stress | De-stress | De-stress |

Table 2: Controlling de-stress ops. using charge pumps.

## 5 INTELLIGENT MEMORY REQUEST SCHEDULER OF HEBE

We design a new memory request scheduling policy to control the aging in the peripheral circuitry within each memory bank using our new aging model (Section 3) and the decoupled peripheral circuit architecture (Section 4).

### 5.1 High-level Overview

We describe HEBE in the context of DRAM-PCM hybrid memory, where embedded DRAM (eDRAM) is used as a write cache to PCM main memory as shown in Figure 6. The baseline memory controller architecture consists of a read-write queue (rwQ) to buffer PCM requests. The key idea of HEBE’s scheduling policy is to 1) improve performance by lowering the de-stress overhead, 2) minimize wasted memory cycles during which a bank is idle with its peripheral circuitry accruing BTI aging, and 3) prevent a request from being delayed too much.

![Figure 6: Request and de-stress scheduling in HEBE.](image)

### 5.2 Detailed Design of HEBE

Figure 6 shows the detailed design of HEBE, which introduces five new components to the baseline memory controller design as highlighted in the figure.

---

*Even though we use eDRAM as cache to PCM in our implementation and evaluations, HEBE is applicable to any type of hybrid memory or standalone PCM memory.*
The first component is the status table (sTab). HEBE uses this table to record if a memory bank is available to serve a PCM request. sTab requires one 1-bit entry for each PCM bank. For 128 banks in a 128GB PCM (see our simulation parameters in Table 3), HEBE requires 128 bits of storage for sTab.

The second component is the access table (aTab). HEBE uses this table to record the number of memory cycles for which a memory bank’s peripheral circuitry is active since the last de-stress operation of the bank. Since peripheral circuitry operates at a different voltage when it is idle than when it is serving a read or a write request, the aging model of HEBE requires the exact number of cycles for which a peripheral circuit is idle and serving read and write requests. Therefore, each aTab entry contains one 16-bit field for recording the idle cycles, and two 4-bit fields for recording the number of reads and write requests. For 128GB PCM with 1GB per bank, HEBE requires 3Kb (= 128 x 24 bits) of storage.

The third component is the unit aging table (uTab). HEBE uses this table to store the unit aging parameters (Eq. 13). Since the three unit aging parameters are the same for every peripheral circuitry in PCM, there are only three 32-bit entries in this table, one for each of these parameters, requiring a total of 96 bits for uTab.5

The fourth component is the request selection. HEBE uses this component to select a request from the rwQ to schedule to PCM. Figure 7 shows the flowchart of HEBE’s request selection mechanism. After scheduling a request from the rwQ, the memory controller checks to see if the number of clock cycles for which a request is outstanding in the rwQ is smaller than the backlogging threshold (\( t_b \)). If the backlogging threshold is exceeded, the request is dequeued and served next. Otherwise, the memory controller selects an outstanding request from the rwQ that is to a bank whose peripheral circuitry has the highest number of idle cycles since the time it served a request from the bank.

The final component is the de-stress selection logic. HEBE uses this component to schedule de-stress operations in PCM banks. For this purpose, HEBE uses two thresholds – the aging threshold (\( t_a \)) and the idle threshold (\( t_i \)). The aging threshold is used to control the aging of peripheral circuitry in PCM in order to achieve a target lifetime. The idle threshold is used to limit the duration during which a peripheral circuit accures aging without doing any useful work. The de-stress selection logic is also shown in Figure 7. If the selected memory request is to a bank whose peripheral circuitry exceeds either of the two thresholds, the memory controller schedules a de-stress operation to the bank. Otherwise, the request is scheduled to PCM.

5For simplicity, we have not considered process variation across different peripheral circuitry of different PCM banks.

5.3 Overhead of HEBE

HEBE requires a total storage of 3.2Kb for a PCM memory of 128GB capacity and 128 banks. The timing overhead of the request and de-stress selection is overlapped with the timing of an ongoing read or a write request, incurring minimal impact on the critical path of PCM read and write accesses. Therefore, HEBE’s request scheduling introduces marginal performance overhead. On the contrary, HEBE improves performance compared to other approaches by reducing the de-stress related performance bottleneck (see Section 7.1).

6 EVALUATION METHODOLOGY

We evaluate HEBE for phase-change memory (PCM), one of the matured NVM technologies. We configure PCM as main memory with eDRAM as its write cache. This is similar to the architecture of IBM POWER 9 [77]. Our simulation framework includes the following components with parameters listed in Table 3.

- Cycle-level in-house x86 multi-core simulator. We configure this to simulate 8 out-of-order cores.
- Main memory simulator, closely matching the JEDEC Non-volatile Dual In-line Memory Module (NVDIMM) specifications [45]. This simulator is composed of Ramulator [41], to simulate DRAM and an cycle-level in-house NVM simulator, based on NVMain [68].
- Power and latency for DRAM and NVM are based on Intel/Micron’s 3D Xpoint specification [73]. Energy is modeled for DRAM using DRAMPower [9] and for NVM using NVMain with parameters from [73].

| Processor | 8 cores, 3 GHz, out-of-order |
|-----------|-----------------------------|
| L1-I/D cache | Private 64KB per core, 4-way |
| L2 cache | Shared, 4MB, 8-way |
| DRAM | 8GB, Micron DDR3 |
| Main Memory | 1 channel, 8 rank/channel, 8 banks/rank, 128 sub-arrays/bank, 512 rows/sub-array |
| PCM | 128GB, Micron DDR3 [73] |
| Main Memory | 2 channels, 1 rank/channel, 8 banks/rank, 64 partitions/bank, 128 tiles/partition, 4096 rows/tile |

Table 3: Major simulation parameters.

Table 4 reports the timing parameters for PCM reads and writes. These parameters are based on Micron’s 128GB PCM module [73].

| Write | 75ns | 15ns | 190ns | 1ns | 209.75ns |
|-------|------|------|-------|-----|----------|

Table 4: PCM timing parameters based on [73].

We evaluate 10 billion instructions of ten workloads from the SPEC CPU2017 benchmarks [7] (see Table 5).

We evaluate the following techniques.
- **Baseline** [84] de-stresses peripheral circuitry of each NVM bank with a fixed \( \text{tdsl} \) of 100, without tracking their aging. Memory requests are scheduled using the FR-FCFS policy [76, 104].
• **HEBE** tracks the aging in CMOS transistors in peripheral circuitry of each bank and de-stresses them only when their aging exceeds the aging threshold. A peripheral circuitry is de-stressed based on the maximum aging of its logic blocks.

• **Decoupled-HEBE** is based on HEBE. Each peripheral circuitry is decoupled to de-stress its logic blocks independently.

| single-core | 8 copies each of blender, bwaves, cactuBSSN, cam4, gcc, mcf, omnetpp, parset, roms, xalancbmk |
|------------|-----------------------------------------------|

Table 5: Evaluated workloads.

6.1 Aging Parameters
To compute aging, the slope parameter of Weibull distribution is set to $\beta = 2$, and the operating temperature is set to 300K. Other fitting parameters are adjusted to achieve an MTTF of 2 years in the baseline system, corresponding to a threshold voltage shift of 10%. This is what is typically accepted as the maximum allowed $V_\text{th}$ degradation before timing errors begin to appear [3, 13–16, 21, 80, 83, 89, 90].

7 RESULTS AND ANALYSES

7.1 Overall System Performance
Figure 8 plots the execution time of each workload for the evaluated systems normalized to Baseline. We make the following two key observations.

First, the average execution time of HEBE is 12% lower than Baseline. This improvement is because HEBE has lower de-stress overhead than Baseline due to HEBE’s dynamic policy to opportunistically de-stress each peripheral circuit only when its aging exceeds the aging threshold. Baseline, on the other hand, uses a fixed de-stress interval of 100 without tracking the exact aging. Second, the average execution time of Decoupled-HEBE is 6% lower than HEBE. This improvement is because Decoupled-HEBE de-stresses the logic blocks in a memory bank’s peripheral circuits off the critical path of read and write accesses from the bank, reducing bank occupancy and improving performance.

![Figure 8: Execution time, normalized to Baseline.](image)

Figure 8: Execution time, normalized to Baseline.

7.2 Overall MTTF
Figure 9 plots the MTTF of each workload for the evaluated systems normalized to Baseline. We make the following two key observations.

First, the average MTTF of HEBE is 16% higher than Baseline. This improvement is because 1) HEBE does not allow the aging of any peripheral circuitry in PCM to exceed the aging threshold and 2) the aging-aware access scheduling policy of HEBE minimizes the number of wasted memory cycles for which a peripheral circuitry accrues aging while being idle. Second, the average MTTF of Decoupled-HEBE is 3.4% higher than HEBE. This improvement is because HEBE needs to wait for an ongoing PCM read or write request to complete before it can schedule the de-stress operation of a peripheral circuitry. Therefore, the circuitry continues to age before it is eventually de-stressed, lowering its MTTF. On the other hand, Decoupled-HEBE can schedule the de-stress operation of a logic block in a bank’s peripheral circuitry independently and in parallel to ongoing read and write requests to the bank. Therefore, the MTTF of Decoupled-HEBE is higher than HEBE.

![Figure 9: MTTF, normalized to Baseline.](image)

Figure 9: MTTF, normalized to Baseline.

7.3 De-stress Overhead
Figure 10 plots the de-stress overhead (Eq. 1) of each workload for each evaluated system normalized to Baseline. We make the following two key observations.

![Figure 10: De-stress overhead, normalized to Baseline.](image)

Figure 10: De-stress overhead, normalized to Baseline.

First, the average de-stress overhead of HEBE is 6.6% lower than Baseline. This improvement is because HEBE increases the de-stress interval ($t_{\text{DSI}}$) by accurately tracking the aging of each peripheral circuitry dynamically, de-stressing it only when aging exceeds a threshold. Baseline uses a fixed $t_{\text{DSI}}$ of 100. Second, the average de-stress overhead of Decoupled-HEBE is 35% lower than HEBE. This improvement is due to the reduction of the de-stress cycle time ($t_{\text{DSC}}$), which is achieved by de-stressing the logic blocks in a bank’s peripheral circuitry independently and in parallel to ongoing read and write requests to the bank.

7.4 Effect of Aging Threshold
Figure 11 reports the execution time and aging of each of our workloads using HEBE, normalized to Baseline. The height of a bar represents HEBE’s result with the default aging threshold of 1000 units. An error bar represents the variation obtained by changing the aging threshold from 500 units to 2000 units. We make the following observation.

![Figure 11: Execution time and aging of HEBE, normalized to Baseline, as a function of the aging threshold.](image)

Figure 11: Execution time and aging of HEBE, normalized to Baseline, as a function of the aging threshold.

When we set a stricter aging threshold (e.g., 500 units), execution time increases and aging decreases. This is because when the...
aging threshold is lowered, the high-voltage exposure time of the peripheral circuitry in each memory bank reduces, reducing the accrued aging. However, performance degrades due to the high de-stress overhead (see Eq. 1). Conversely, when we relax the aging threshold (e.g., 2000 units), the de-stress interval increases, reducing the de-stress overhead and increasing performance. However, aging is now higher because of longer exposure to high-voltage stress.

7.5 Temperature Dependency
Figure 12 plots MTTF of Decoupled-HEBE at 300K, 325K, and 350K normalized to Baseline at 300K for each evaluated application. We observe that MTTF decreases with increase in temperature. MTTF at 325K and 350K are higher than at 300K by an average of 7% and 26%, respectively. These results follow directly from our aging formulation, which incorporates temperature using the scaling parameter $\alpha$ in Eq. 5. This parameter grows exponentially with temperature, resulting in a corresponding exponential increase in aging. More aging leads to larger shift in threshold voltage.

Figure 12: MTTF at 325K and 350K normalized to Baseline.

8 RELATED WORKS
To our knowledge, this is the first work that exploits a workload’s access characteristics to dynamically control the length of the de-stress interval of peripheral circuitry in each memory bank, improving both performance and MTTF of PCM-based main memory. Many works propose optimizations for PCM. Recent examples include architecture optimization [47, 70, 97], performance and energy optimization [1, 37, 81, 84, 99, 100], wear leveling [85, 101, 102], and memory controller optimizations [103]. See [95] for a survey of these and other similar approaches. HEBE can be combined with most of these techniques.

Many works propose memory latency reduction, refresh optimization, energy reduction, and request scheduling methods to enhance system performance, fairness, quality of service, or security [2, 10–12, 23, 24, 27, 30, 32–35, 38, 39, 50–52, 55, 62, 63, 65–67, 72, 74–76, 79, 87, 88, 91]. None of these works consider aging of phase change memory in their scheduling decisions. Our aging-aware scheduling mechanism can be incorporated into other memory controller designs that aim to improve other metrics.

9 CONCLUSIONS
We introduce HEBE, a new mechanism that can dynamically track and control the aging of transistors in peripheral circuitry of each memory bank, improving both performance and aging of NVM-based main memory. HEBE is built on three novel contributions. First, we propose a new, accurate analytical model to dynamically track aging in response to the memory controller’s request scheduling decisions. Second, we develop a new, intelligent request scheduler that exploits this aging model at run time to decide when peripheral circuitry in NVM must be de-stressed. Third, we decouple logic blocks in peripheral circuitry operating at different voltages, allowing these blocks to be de-stressed independently and off the critical path of execution, improving performance. We evaluate HEBE for DRAM-NVM hybrid main memory and show the significant performance and MTTF improvement. We conclude that HEBE is a simple yet powerful mechanism to dynamically manage the aging in non-volatile main memory, and improve both performance and lifetime via its intelligent request scheduling decisions.

ACKNOWLEDGMENTS
This work is supported by the National Science Foundation Faculty Early Career Development Award CCF-1942697 (CAREER: Facilitating Dependable Neuromorphic Computing: Vision, Architecture, and Impact on Programmability).

REFERENCES
[1] M. Arjomand et al., “Boosting access parallelism to PCM-based main memory,” in ISCA, 2016.
[2] R. Ausavarungnirun et al., “Staged memory scheduling: Achieving high performance and scalability in heterogeneous systems,” in ISCA, 2012.
[3] A. Balaji et al., “A framework to explore workload-specific performance and lifetime trade-offs in neuromorphic computing,” CAL, 2019.
[4] A. Balaji et al., “Mapping spiking neural networks to neuromorphic hardware,” TVLSI, 2020.
[5] A. Balaji et al., “Enabling resource-aware mapping of spiking neural networks via spatial decomposition,” ESL, 2020.
[6] C. Bolchini et al., “A lightweight and open-source framework for the lifetime estimation of multicore systems,” in ICCD, 2014.
[7] J. Bucek et al., “SPEC CPU2017: Next-generation compute benchmark,” in JCP, 2018.
[8] G. W. Burr et al., “Neuromorphic computing using non-volatile memory,” Advances in Physics X, 2017.
[9] K. Chandrasekar et al., “DRAMPower: Open-source DRAM power & energy estimation tool,” URL: http://www.drampower.info, 2012.
[10] K. K. Chang et al., “Understanding latency variation in modern DRAM chips: Experimental characterization, analysis, and optimization,” in SIGMETRICS, 2016.
[11] K. K. Chang et al., “Understanding reduced-voltage operation in modern DRAM devises: Experimental characterization, analysis, and mechanisms,” in SIGMETRICS, 2017.
[12] K. K.-W. Chang et al., “Improving DRAM performance by parallelizing refreshes with accesses,” in HPCA, 2014.
[13] A. Das et al., “Fault-aware task re-mapping for throughput constrained multimedia applications on NoC-based MPSoCs,” in RSP, 2012.
[14] A. Das et al., “Fault-tolerant network interface for spatial division multiplexing-based network-on-chip,” in ReCoSoC, 2012.
[15] A. Das et al., “Aging-aware hardware-software task partitioning for reliable reconfigurable multiprocessor systems,” in CASES, 2013.
[16] A. Das et al., “Energy-aware dynamic reconfiguration of communication-centric applications for reliable MPSoCs,” in ReCoSoC, 2013.
[17] A. Das et al., “Energy-aware task mapping and scheduling for reliable embedded computing systems,” in TETCS, 2014.
[18] A. Das et al., “Temperature aware energy-reliability trade-offs for mapping of throughput-constrained applications on multimedia MPSoCs,” in DATE, 2014.
[19] A. Das et al., “Combined DVFS and mapping exploration for lifetime and soft-error susceptibility improvement in MPSoCs,” in DATE, 2014.
[20] A. Das et al., “Reinforcement learning-based inter-and intra-application thermal optimization for lifetime improvement of multicore systems,” in DAC, 2014.
[21] A. Das et al., “Reliability and energy-aware mapping and scheduling of multimedia applications on multiprocessor systems,” in TPDS, 2015.
[22] A. Das et al., “VRD-DRAM: Improving DRAM performance via variable refresh latency,” in DAC, 2018.
[23] H. David et al., “Memory power management via dynamic voltage/frequency scaling,” in IAC, 2011.
[24] Q. Deng et al., “MemScale: Active low-power modes for main memory,” in ASPLOS, 2011.
