In Proceedings of the 17th IEEE International Symposium on High Performance Computer Architecture (HPCA), 2011

Hardware/Software Techniques for DRAM Thermal Management

Song Liu, Brian Leung, Alexander Neckar†[1], Seda Ogrenci Memik, Gokhan Memik, Nikos Hardavellas
Northwestern University, †Stanford University
{sli646, ble973, seda, memik}@eecs.northwestern.edu, nikos@northwestern.edu, aneckar@stanford.edu

Abstract

The performance of the main memory is an important factor on overall system performance. To improve DRAM performance, designers have been increasing chip densities and the number of memory modules. However, these approaches increase power consumption and operating temperatures: temperatures in existing DRAM modules can rise to over 95°C. Another important property of DRAM temperature is the large variation in DRAM chip temperatures. In this paper, we present our analysis collected from measurements on a real system indicating that temperatures across DRAM chips can vary by over 10°C. This work aims to minimize this variation as well as the peak DRAM temperature. We first develop a thermal model to estimate the temperature of DRAM chips and validate this model against real temperature measurements. We then propose three hardware and software schemes to reduce peak temperatures. The first technique introduces a new cache line replacement policy that reduces the number of accesses to the overheating DRAM chips. The second technique utilizes a Memory Write Buffer to improve the access efficiency of the overheated chips. The third scheme intelligently allocates pages to relatively cooler ranks of the DIMM. Our experiments show that in a high performance memory system, our schemes reduce the peak DRAM chip temperature by as much as 8.39°C over 10 workloads (5.36°C on average). Our schemes also improve performance mainly due to reduction in thermal emergencies: for a baseline system with memory bandwidth throttling scheme, the IPC is improved by as much as 15.8% (4.1% on average).

1. Introduction

Major microprocessor manufacturers are adopting multi-core designs to take advantage of hardware parallelism and large scale integration [2-5]. The success of this multi-core era hinges upon the assumption of a comparable increase in memory subsystem performance. Thus, DRAM designers often increase main memory bandwidths and chip densities to improve performance. However, these methods exacerbate the problems of high power consumption and operating temperatures in DRAM systems. Experiments on several thin-and-light laptops show that 1GB SO-DIMMs reach the maximum case temperature rating of 85°C and sometimes exceed this limit [6]. In servers, the temperature of the DRAM can exceed 95°C [1]. These high temperatures have adverse effects on the performance and reliability of the DRAM. When the temperature of the DRAM reaches a critical level, the memory traffic has to be throttled down. Intel Centrino Duo implements two throttling techniques to control the DRAM temperatures [6]. The first, Delta Temperature in Serial Presence Detect (DT-in-SPD), throttles the DRAM according to temperature predictions. The other method throttles memory traffic according to the readings from a temperature sensor on the memory module. Note that both solutions result in decreased performance. In Section 6, we show that our schemes can improve the overall system performance by up to 15.8% when compared to such a bandwidth throttling scheme. Elevated temperatures also adversely affect reliability: according to the Arrhenius equation [7], a 10°C difference in temperature roughly halves the Mean Time Between Failure [7-8].

DRAM thermal problems are further exacerbated as designers squeeze more DRAM chips on the already densely populated DIMMs. The physical constraints place limitations on the DIMM pitch and allowable thermal solutions. Furthermore, the dense and uniform placement of memory devices on the DIMM makes it challenging to enable an effective spreading of generated heat. As a result, heat tends to accumulate at a faster rate within the crowded DIMM [9].

Figure 1 illustrates an example DIMM layout in high performance servers [10]. For illustrative purposes, we depict a generic DIMM, which contains a number of DRAM devices and a buffer chip. Such DIMMs are referred to as buffered DIMMs, which are widely used in servers. However, the basic layouts of other memory modules (buffered or unbuffered types) are similar and the issues pertaining to thermal behavior apply to all of them. In this figure, four DIMMs are placed in a row. Processors are placed at the upwind side of the DIMMs. Cooling airflow first passes over the processors and then proceeds along channels between DIMMs. Hence, the system architecture greatly influences the thermal behavior of the DRAM chips. Downstream chips tend to get considerably warmer.

Previous studies have measured thermal profiles of high performance memory modules [1, 9]. Figure 2

[1] Alexander Neckar was affiliated with Northwestern University when this work was performed.

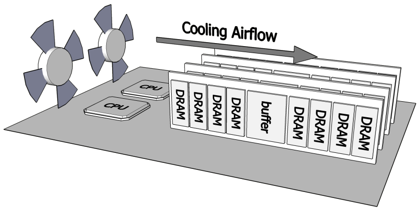


Figure 1. DIMM in the system.

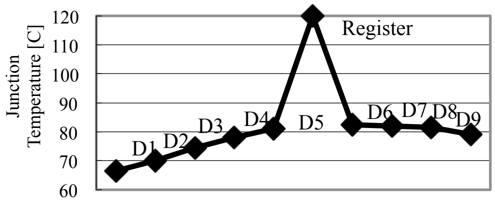


Figure 2. Typical temperature profile along the RDIMM [1].

shows the typical thermal profile in a Registered DIMM (RDIMM) as reported by Zhu et al. [1]. Note that although the register chip is the hottest component, it may not cause a bottleneck; DRAM chips and buffer chips are rated separately at different maximum temperatures, and DRAM chips typically have much lower critical temperature thresholds. An important observation in Figure 2 is that the temperature difference between the hottest DRAM chip (D6) and the coolest DRAM chip (D1) is over 15°C. We observe similar trends in our experiments with a desktop system described in Section 3. For large temperature variations, current solutions throttle the whole DIMM even when the coolest chip (and often several other chips on the DIMM) is far below its maximum temperature specification. The main reason for these temperature fluctuations is the flow of cool air—DRAM chips in a rank are typically accessed in parallel and consume an identical amount of power. Our thermal-aware architecture takes into account the existence of this cooling variation to reduce temperature variation. Specifically, we present a novel ***hot/cool rank organization*** that enables independent DRAM accesses to hot and cool DRAM chips that leverages cooling variations for thermal benefits. We present temperature-aware DRAM architectures that reduce DRAM temperature variations and peak temperature by changing the DRAM access patterns. Specifically, we propose a new cache line replacement policy in the last level of cache, ***Temperature Aware Least Recently Used (TA-LRU)***, that tries to evict cache lines from cooler DRAM chips. We also propose a second technique, ***Temperature Aware Memory Write Buffer (TA-MWB)***, to reduce DRAM power consumption of overheated DRAM chips. This is achieved by intelligently stalling write operations to improve the row buffer hit rate for the hot DRAM chips. Finally, we explore an OS-level scheme, ***Temperature-Aware Page Allocation scheme (TA-PA)***, to better distribute heat throughout the DIMM.

We evaluate our proposed techniques on a CMP system with four processor cores running SPEC 2000 and 2006, MineBench, BioBench, web server, and database applications. Experiments show that our proposed temperature aware architectural optimizations reduce the peak DRAM steady-state chip temperature by as much as 8.39°C (5.36°C on average) over 10 representative workloads on 2 different baseline systems (LRU and clock-based pseudo LRU). For the five workloads that are experiencing DRAM thermal throttling, this reduction in temperature yields up to 10.9% DRAM bandwidth improvement (6.4% on average), which results in up to 15.8% improvement in system performance (4.1% on average). For applications not causing any thermal emergencies, the penalty of our approaches remain negligible.

The remainder of the paper is organized as follows. Section 2 reviews related work in DRAM power and temperature management. Section 3 presents our experiments analyzing DRAM chip temperatures. Section 4 presents our DIMM thermal model. Our proposed DRAM thermal management techniques are presented in Section 5. After discussing the experimental results in Section 6, we conclude our work in Section 7.

2. Related Work

Traditionally, DRAM management has two main optimization goals: maximizing performance or maximizing a combined energy-performance metric, e.g., energy-delay product. In high performance systems, the memory space distribution among ranks/banks is interleaved. The benefit of interleaved organization is to distribute DRAM accesses evenly across all ranks for maximum bandwidth and thus, best system performance. Another approach involves restricting DRAM accesses for a certain duration of time to a subset of DRAM banks so that other banks can be in low power states to conserve power [11-15].

Recent studies show that DRAM thermal management is a pressing issue in high performance mobile systems and servers [1, 6, 9]. Two particular technologies deployed on mobile systems are the on-DIMM thermal sensors and DT-in-SPD. Guided by these monitoring schemes, dynamic thermal management (DTM) techniques are used to control the DRAM activity, and hence, temperature. Lin et al. [10, 15] propose adaptive core gating and DVFS for CMP systems similar to those used for processors and disk drives. However, such techniques are known to introduce system performance penalties, because their end effect is a direct reduction of DRAM accesses available per unit time. Our previous work [16-17] inserts a buffer to improve page hit rate. This in turn reduces power consumption and thermal dissipation.

The main drawback in all of the above mentioned techniques is that they overlook cooling variations within the DRAM system. As a consequence, these management techniques lead to overly aggressive

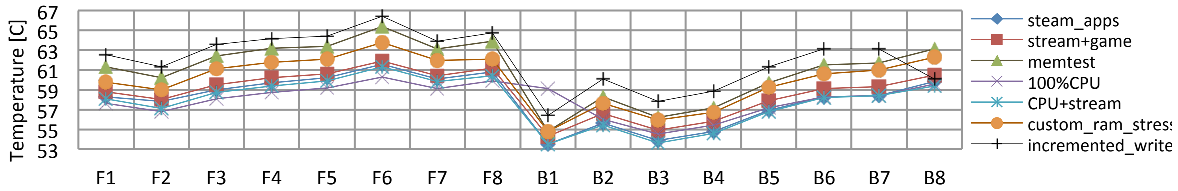


Figure 3. Temperature variation of DRAM chips across a DIMM on a desktop machine. F1-F8 are the chips on the front of the DIMM whereas B1-B8 are the chips on the back of the DIMM.

throttling of the memory system, which leads to the underutilization of the thermal budget of the system. Our techniques differ in that they minimize the temperature variation that exists across the DIMM and operate at a much finer granularity. In addition, we propose changes to the caching mechanism of the CPU to enhance the DRAM thermal behavior.

Our proposed techniques are orthogonal to dynamic thermal management (DTM) techniques. Our goal in this paper is to reduce the number of thermal emergencies by leveraging DRAM cooling variations whereas various DTM techniques are proposed to handle thermal emergencies efficiently. Our proposed techniques could easily improve such DTM techniques. In section 6.3, we demonstrate the effectiveness of our methods on a system that has DRAM throttling.

3. Temperature Measurements

We attached thermal sensors to the DRAM chips (DDR2) inside a Dell Optiplex 760 Desktop to obtain temperature measurements. In order to stabilize the ambient temperature to 45°C during our tests, the desktop is placed inside an enclosure with a variable temperature controller. Figure 3 shows the temperature variations for the DRAM chips on the DIMM for different applications that we used to stress the system. We observe that the temperature range across the DRAM chips on a DIMM can vary by up to 10.6°C. Our experiments confirm that there is a large variation among individual DRAM chip temperatures. The lower absolute temperatures (compared to server temperatures) are due to the relatively lower performance desktop used in the experiments.

4. Thermal Model of the DIMM

According to the thermal behavior, we categorize DIMMs into two types: buffered DIMMs and unbuffered DIMMs. Buffered DIMMs, such as the RDIMM and the Fully Buffered DIMM (FB-DIMM), are typically used in high performance systems. They contain an extra chip for buffering data. We refer to this additional chip as the buffer chip. On the other hand, unbuffered DIMMs do not have buffer chips. In both FB-DIMM and RDIMM, the buffer chip has much higher power consumption than the DRAM chips. Therefore the presence of the buffer chip has a strong impact on the thermal profile of the DIMM. In this section, we describe our detailed thermal model for a DIMM that can be easily adapted to capture both types of DIMMs.

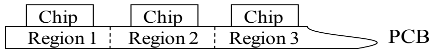


Figure 4. Splitting the PCB into regions in the DIMM thermal model.

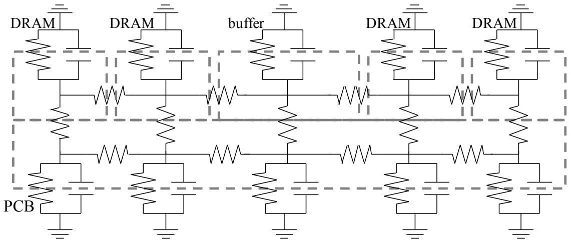


Figure 5. Thermal model for the buffered DIMM.

Before building the thermal model of the DIMM, we make several reasonable simplifications. First, we ignore the thermal interactions between the DIMM and certain components in the system (e.g., CPU). The impact of CPU temperature is reflected by the ambient temperature. Second, the thermal resistance on the direction perpendicular to the Printed Circuit Board (PCB) is much larger than the thermal resistance on the direction parallel to the surface of the PCB; hence, we ignore thermal interaction between chips on two sides of the DIMM. Therefore, our DIMM thermal model focuses on the chips on one side of the DIMM.

We develop our model following the basic principles of the thermal RC model [15, 18]. In this model, we consider each chip on one side of the DIMM and the PCB. As shown in Figure 4, the PCB is split into several regions. There are thermal resistances between each component and the ambient environment. Neighboring chips have thermal interactions with each other. Due to cooling airflow, thermal resistances between neighboring chips on the downwind direction are smaller than those on the upwind direction. Chips on the DIMM also interact through the PCB. We assume that thermal resistances in the PCB are same in the direction of the airflow. Figure 5 demonstrates our proposed thermal model for the buffered DIMM. For simplicity, only four DRAM chips are illustrated.

5. Temperature Aware DRAM Management Schemes

In this section, we first discuss DIMM organization that enables independent DRAM traffic to hot and cool DRAM chips on a DIMM. Then, we describe our proposed techniques, TA-LRU, TA-MWB, and TA-PA for temperature aware DRAM management.

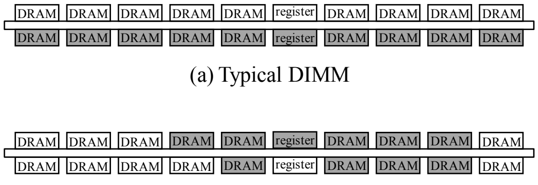
5.1 Hot/Cool Rank Organization

In many high performance servers, the DIMM is divided into 2 ranks as shown in Figure 6(a). The white chips belong to the first rank, and the gray chips belong to the second rank. As previously shown in Figure 2 and Figure 3, the DRAM chip temperatures vary along a DIMM and the DRAM chips neighboring the register have the worst cooling conditions. We take advantage of this cooling variation for thermal benefits. We propose an alternative DIMM layout that enables independent accesses to hotter and cooler DRAM chips; particularly, we statically categorize chips into a hot and a cool rank. As shown in Figure 6(b), one rank consists of all the chips that are expected to be hot (indicated with gray in the figure), and the other rank consists of all the DRAM chips that are expected to be cool (white chips in the figure).

All the hardware/software schemes that we develop utilize this new rank distribution. The basis of our optimizations is contingent upon how the two ranks are organized to better distribute the heat so that throttling can be minimized since chips are less likely to overheat. By generating uneven DRAM traffic to the hot and cool ranks, we reduce the DRAM temperature variation and prevent a single chip from overheating (and hence causing throttling).

5.2. Temperature Aware LRU Cache

In a set associative cache, each address is mapped into a set that contains multiple cache lines. When a new cache line is entered into the cache, one of the existing cache lines has to be evicted, typically with the Least-Recently Used (LRU) replacement policy. In LRU, the cache line that has not been accessed for the longest duration of time is evicted from the cache. Many techniques have been proposed to optimize the system by changing the cache replacement policy [8, 19-20]. In this section, we propose a Temperature Aware LRU (TA-LRU) policy for DRAM temperature management. Note that changing the cache replacement policy may incur performance penalty. Our experiments, described in Section 6, however, show that the performance penalty is negligible.



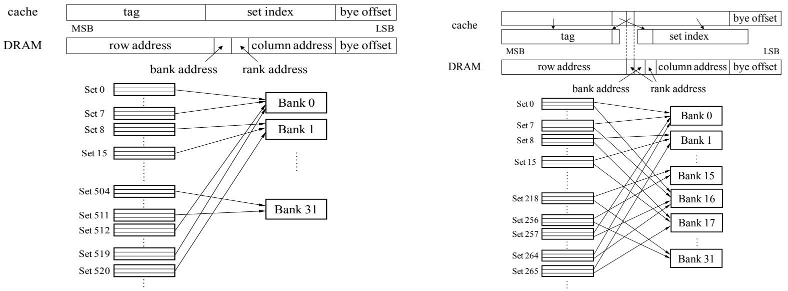
(a) Typical DIMM

(b) Hot/cool separated DIMM

Figure 6. Rank distribution of chips in a RDIMM with 2 ranks: a) a typical distribution and b) our proposed hot/cool rank separation.

Our TA-LRU policy keeps cache lines from the hot rank longer in the cache and reduces the number of future accesses to the hot rank. Before applying a new cache line replacement policy, we change the organization of the last level cache so that data from different ranks are mapped into the same set of cache entries. As shown in Figure 7(a), an address used to access the cache is decoded as byte offset, set index, and tag (note that the lower level caches are physically indexed, which is the same in our approach). In the DRAM system, the address is decoded as byte offset, rank address, bank address, and row address. In most systems, the last-level cache has thousands of sets; hence, the bank address and rank address are part of the set address. Therefore, cache entries in one set will be mapped to only a single rank. In order to utilize our proposed cache replacement policy, we change the memory decoding in the last level of the cache as shown in Figure 7(b). By changing the mapping of bank addresses, the most significant bit of the bank/rank address becomes part of the cache tag. With this configuration, each cache set may have data from two DRAM ranks. These two DRAM ranks are called to be ***paired***. For convenience, we put the hot rank and the cool rank within one DIMM in a pair. The hot/cool designation is stored on the DIMM and provided to the processor during system boots (similar to DT-in-SPD).

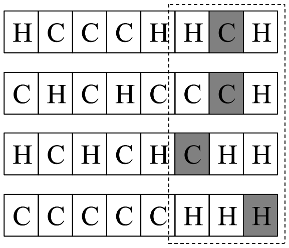
To balance the temperature in the DIMM, we give higher priority of utilizing the cache for those cache lines originating from the hotter DRAM ranks. Providing excessive priority to data from the hotter DRAM ranks will hurt the efficiency of the cache. Our proposed temperature aware LRU policy works as



(a) System with state-of-the-art cache

(b) System with TA-LRU cache

Figure 7. Memory decoding in the baseline cache architecture and TA-LRU cache.



H	C	C	C	H	H	C	H
C	H	C	H	C	C	C	H
H	C	H	C	H	C	H	H
C	C	C	C	C	H	H	H

Figure 8. Cache line replacement example with *m* = 3.

follows. For an *n*-way set associative cache, we check the source of the *m* least recently used entries. If one or more cache lines in these entries are from the cooler DRAM ranks, we replace the least recently used cache line among those associated with these cooler ranks. When all the *m* cache lines are from ranks, the least recently used cache line will be discarded.

Figure 8 shows an example for the cache line replacement in an 8-way associative cache with $m = 3$. "H" denotes a cache line from a hot DRAM rank; "C" denotes a cache line from a cooler rank. Most recently used cache lines are placed on the left-hand side. Three least recently used cache lines in each set are enclosed in the dashed rectangle. The shadowed block in each row denotes the cache line to be discarded. In the first three examples, there is at least one cache line within the rectangle from a cold DRAM rank. Therefore, the least recently used "cold" cache line is discarded. In the last case, all three least recently used cache lines are from the hot DRAM ranks, hence, the least recently used cache line is discarded.

In TA-LRU, *m* is an important parameter to balance the tradeoff between higher performance and more aggressive thermal management. A larger *m* results in a stronger impact on power distribution among different DRAM ranks at the expense of a higher performance penalty. When $m = 1$, TA-LRU becomes the traditional LRU policy. When $m = n$, almost the entire cache is reserved for data from overheated DRAM ranks. Our approach to determine m was to use a sensitivity analysis to assess the impact of *m* on the system performance and DRAM temperature variation. For an 8-way associative cache, the optimal configuration is achieved when $m = 3$.

5.2.1 Temperature Aware Pseudo-LRU Cache

In most modern processors, the last level cache does not utilize a traditional LRU replacement policy. Instead, several pseudo-LRU policies [21] are developed. Some frequently used pseudo-LRU policies are Tree-based Pseudo-LRU (TPLRU), Clock-Based LRU (CLRU), and Not-Most-Recently-Used (NMRU). Temperature aware features could also be enabled in these pseudo-LRU policies. Detailed descriptions of each policy are omitted due to space; we show the experimental results for LRU and CLRU in Section 6.

5.3. Temperature Aware Memory Write Buffer

In Temperature-Aware Memory Write Buffer (TA-MWB), we extend the memory controller to include a buffer for storing the write operations. The idea behind this buffer is to delay the write operations to the memory and thus, increase the efficiency of the read operations. Since the write operations are not on the critical path of the execution, we can delay them and consecutive read operations can hit the open-page of the DRAM. This, in return, improves the performance of the memory and reduces its power consumption.

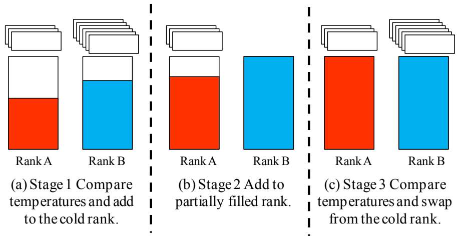

Rank A
Rank B
(a) Stage 1 Compare temperatures and add to the cold rank.
Rank A
Rank B
(b) Stage 2 Add to partially filled rank.
Rank A
Rank B
(c) Stage 3 Compare temperatures and swap from the cold rank.

Figure 9. Three stages in the TA-PA scheme.

The idea behind the TA-MWB is similar to the idea behind the Page Hit Aware Write Buffer (PHA-WB) in our previous work [16-17]. The PHA-WB focuses on the improvement of DRAM power efficiency for all DRAM chips; while our proposed TA-MWB is tailored towards optimizing the power efficiency of DRAM chips under thermal stress. Instead of buffering write operations to all the DRAM chips evenly (as the PHA-WB does), write operations to the hot DRAM chips have higher priority and remain in the buffer longer. Similar to the TA-LRU scheme, we manage an *n*-entry write buffer as a set of cache lines. When the write buffer is full and another operation needs buffering, one operation in the buffer will access the DRAM. If one or more of the *m* oldest operations in the buffer is accessing a cool rank, the oldest is executed. If all the *m* oldest operations access a hot chip, we select the oldest among them to execute.

5.4. Temperature Aware Page Allocation

In this section, we detail our Temperature-Aware Page Allocation (TA-PA) scheme, which operates on the operating system (OS) level. The main goal of the TA-PA scheme is to reduce the page allocation to the hotter ranks (accesses to them) while increasing the allocation on the colder ranks. In the following, we describe our scheme for a two-rank system without loss of generality. In TA-PA, there are three possible stages (as shown in Figure 9) at any point in time:

(a) Rank A and B are not full. Every epoch, we check which rank is cooler by comparing the maximum temperature on both ranks. The following pages are then allocated to the cooler rank until the next temperature estimation cycle.

(b) One rank is not full, and the other rank is full. New pages are added to the partially filled rank.

(c) Rank A and B are both full. Every epoch, we compare the maximum temperature on both ranks. Then, we make page swaps to the cooler rank until the next temperature estimation cycle.

This algorithm requires several parameters. First, it needs information about the rank temperatures, which could either be measured using sensors or predicted using a model like ours. Second, it requires the execution granularity: during our experiments, the algorithm is run every 1 million cycles.

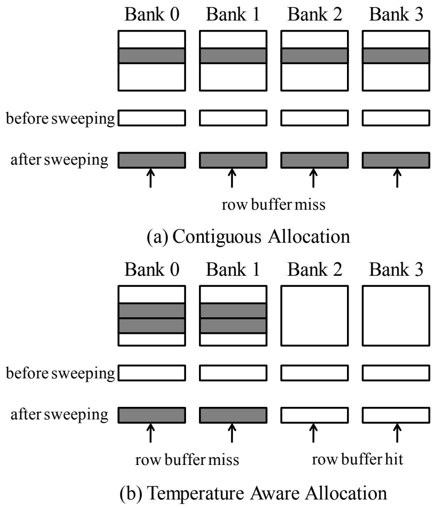
An important property of our algorithm is that it may make more allocations on one rank compared to the other. To limit this skewed allocation, we use a threshold restricting the difference between the numbers of pages allocated to either rank. In our experiments, we use a ratio of 1 to 10; in other words, if the difference in allocations exceeds a 1-to-10 ratio, we allocate pages to the hot rank. We must also note that the allocation of pages will be highly skewed if the temperature of one rank remains cold despite the allocations. This is possible only if a) the overall activity on the memory is low or b) allocating more pages to a rank does not result in increased accesses to it. In either case, the performance impact of allocating more pages to one rank should remain low. In fact, as we discuss further in this section and in Section 6, TA-PA results in overall performance improvement of the system for a large majority of the workloads we study.

Please note that our TA-PA scheme reduces temperature variation along the DIMM during Stage 1 and Stage 3 and increases temperature variation in Stage 2 (during which we may allocate more pages to the hot rank). However, Stage 2 occupies a short interval of time in applications with high memory traffic. Our experiments show that, among the 37 tested workloads, only one of them spends a relatively long duration in Stage 2 (about 50% of the simulation time). Other workloads either reach Stage 3 in less than 2 billion cycles (12 workloads), or spend their whole execution in Stage 1 (24 workloads). This is not surprising as a) only workloads with large memory footprints will fill both ranks and b) for those workloads with larger footprints, Stage 2 will be an intermediate stage; the only workloads that will spend a long duration in Stage 2 are the ones that have the footprint that just fills one rank and leaves the other partially filled.

5.4.1. Performance Impact of TA-PA

In most systems, pages are interleaved in different memory banks, meaning contiguous pages are placed in different banks. When allocating a large piece of memory, it is preferable to use contiguous pages in the physical memory for higher bank-level parallelism. However, due to fragmentation problems, allocating contiguous pages is not always possible. Modern operating systems provide noncontiguous memory allocation, where noncontiguous physical memory can be used as contiguous virtual memory. Our TA-PA scheme falls into this category. Our proposed technique prefers noncontiguous physical memory that allocates a large object in a subset of all the DRAM banks for DRAM thermal benefits.

Generally, it is believed that allocating large objects in contiguous physical memory would yield better performance. However, this is not necessarily the case. The following example demonstrates a scenario that prefers noncontiguous allocation. Figure 10 shows a system with 4 memory banks. In this example, we consider the allocation of an array that fills four rows. In Figure 10(a), the array is placed in contiguous memory that covers all the four banks (as shown with gray blocks). Consider what happens when the application sweeps through this array. After the sweeping, this array fills the row buffers of all the memory banks. Any accesses to previous contents in the row buffer will cause a row buffer miss. We refer to this phenomenon as "row buffer pollution." Temperature-aware noncontiguous allocation, on the other hand, reduces the effect of "row buffer pollution." As shown in Figure 10(b), when the same sweeping operation is carried out on the TA-PA scheme, only two banks have their row buffers polluted. Future accesses to the other two banks may enjoy a row buffer hit. As a result, some workloads may actually see an improved performance; in fact, we have observed that our TA-PA improves the performance by up to 18.2%.



(a) Contiguous Allocation

(b) Temperature Aware Allocation

Figure 10. Example access pattern of row buffer pollution, where temperature aware noncontiguous allocation outperforms contiguous allocation.

Utilizing noncontiguous allocation also impacts available bank level parallelism. Noncontiguous allocation could either improve the bank level parallelism by moving back-to-back memory accesses from one bank to different banks or hurt bank level parallelism by moving accesses to multiple banks into one. Our experiments show that on average our technique has positive impact on system performance.

TA-PA also requires minor changes to the page management of the operating system. In Stage 3, instead of managing all the pages in a large cyclic list, the OS manages pages in different ranks separately. This does not incur any overhead in the OS since the total number of pages does not change. Changing the page management may also change the number of page swaps. Our experiments show that TA-PA reduces the number of page swaps by 7% over 10 workloads.

6. Experimental Results

In this section, we first present our experimental

Table 1. Experimental configuration of processor, memory system, and temperature aware DRAM management techniques.

Parameters	Values
Processor I-cache (per core) D-Cache (per core) L2 Cache (shared) L2 prefetch units	4-core, 3GHz 32KB, 8-way, 64B line, 2 cycle hit latency 32KB, 8-way, 64B line, 2 cycle hit latency, write-back 2MB, 8-way, 64B line, 10 cycle hit latency, write-back 256 entry IP-based prefetch unit and 12 entry stream-based prefetch unit
FB-DIMM DDR2 DRAM Chip Burst Mode Major Timing Parameter	1GB, 512Mb per chip, 8 DQs per chip, 800MHz 4 banks per chip, 16384 rows per bank, 256 columns per row, 4 bytes per column Burst length of 8 Active to Read tRCD=12.5ns, Read to Read Data tCL=12.5ns, Precharge to Active tRP=12.5ns
TA-LRU TA-MWB	*m* = 3 64 entries, *m* = 32

setup, including the power models, simulator configuration, and experimental methodology. Then, we describe the results from our simulations.

6.1. Experimental Setup

We use a DRAM power model based on the data published by Micron Technology Inc. for DDR, DDR2, and DDR3 memory chips [22]. For our simulations, we model an FB-DIMM device. The AMB power model used in our experiments is developed by Lin et al. [15]. We use the Zesto simulator [23] as our architectural simulator to evaluate our techniques. We implement the FB-DIMM power model and our thermal model in Zesto. We simulate a CMP with four processor cores and 1 GB DDR2 FB-DIMM. In our simulations, each core executes one application. The major parameters for the processor and memory are listed in Table 1. We simulate the TA-LRU scheme, the TA-MWB scheme, a combination of the two mentioned techniques, and the TA-PA scheme.

We simulate the OS behavior in the TA-PA scheme by remapping memory addresses. Specifically, we monitor the first access to a page and allocate an address to it in the physical address in the DRAM (based on the current stage of the TA-PA and availability). Once such an address is generated for a page, all accesses to that page are remapped accordingly. When the memory is full, we apply the page replacement methodology (the base case and the TA-PA) to find the page to be swapped.

We tested 40 applications from the SPEC CPU 2000 benchmarks [24], SPEC CPU 2006 benchmarks [24], MineBench [25], and BioBench [26]. Among these benchmarks, we selected five applications—one with the highest IPC (*tigr* from BioBench), two with the highest number of L2 cache accesses (*soplex* from SPEC CPU 2006, and *vortex* from SPEC CPU 2000), and two with the heaviest DRAM traffic (*swim* from SPEC CPU 2000 and *ECLAT* from MineBench). We then constructed ten workloads using these five applications. The first five workloads each have four instances of these selected single applications. In order to avoid identical L2 cache accesses from these four processor cores, we fast forward a different number of instructions for these four instances. We fast forward 100M instructions for the first one, 200M for the second one, and so on. The other five workloads have four different applications (all possible combinations of the five selected applications). The workload contents are described in Table 2. Table 3 contains the access related parameters for these workloads. We present the total IPC over 4 cores, the number of DRAM accesses, the L2 cache miss rates for all baseline systems and their temperature aware correspondents, the maximum steady-state temperature observed among the DRAM chips on the DIMM, and the steady-state AMB temperature. Note that, our system is enabled with L2 cache prefetch units. Therefore, the number of DRAM accesses is greater than the number of L2 cache lookups multiplied by the cache miss rate. We simulate each workload for 1 billion cycles. We track the number of accesses to each DRAM chip and the row buffer hit rate on each chip. These statistics are used by our thermal model to calculate the temperature for each chip.

In addition to the workloads mentioned above, we consider the commercial workloads used in data centers to evaluate our TA-PA. More specifically, we use two web servers and two online transaction processing databases. These data intensive workloads have large footprints and stress the DRAM by creating heavy DRAM traffic. We evaluate our TA-PA on these workloads with traces of memory accesses collected using FLEXUS [27]. We simulate a multi-core processor with 16 cores supported by 64KB 2-way set-associative split I/D L1 caches, a 16MB 16-way set-associative L2 cache with 64B cache blocks, and a 16GB DRAM (4 DIMMs). The cores implement the UltraSPARC III ISA and execute unmodified server applications running on a Solaris 8 operating system. Parameters of the server workloads are in Table 4.

6.2. Architectural Results

Our proposed techniques impact the system performance in two different ways. First, our techniques may impact the cache hit rate, DRAM row buffer hit rate, and bank level parallelism. We show such effects in Section 6.2. In summary, our results indicate that our proposed techniques have a negligible performance penalty on applications that are not experiencing thermal emergencies. Second, our techniques reduce temperature variation and thus, the

Table 2. Workload mixes created of applications from SPEC, BioBench, and MineBench.

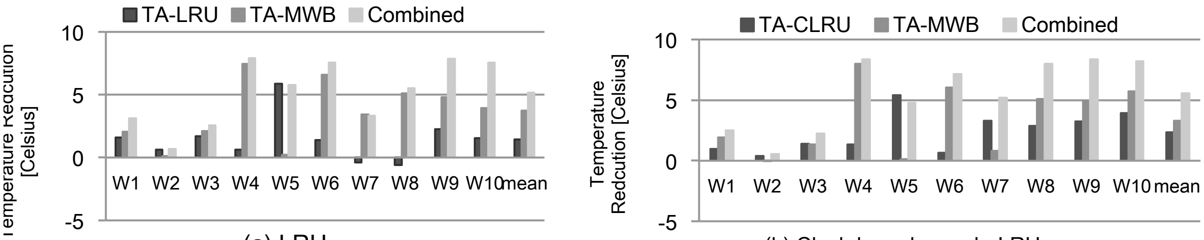
Workload	Benchmarks
W1	tigr, tigr, tigr, tigr
W2	soplex, soplex, soplex, soplex
W3	vortex, vortex, vortex, vortex
W4	swim, swim, swim, swim
W5	ECLAT, ECLAT, ECLAT, ECLAT
W6	tigr, soplex, vortex, swim
W7	tigr, soplex, vortex, ECLAT
W8	tigr, soplex, swim, ECLAT
W9	tigr, vortex, swim, ELCAT
W10	soplex, vortex, swim, ECLAT

Table 4. Data intensive commercial workloads. These applications are known to have heavy DRAM traffic.

OLTP – Online Transaction Processing (TPC-C v3.0)	
iosrv, thru4, qry 2, 6, 8, 13, 13, 16,17	*IBM DB2 v8 ESE* 100 warehouses (10 GB), 64 clients, 2 GB buffer pool
oracle	*10g Enterprise Database Server* 100 warehouses (10 GB), 16 clients, 1.4 GB SGA
Web Server	
apache	*Apache HTTP Server v2.0* 16K connections, fastCGI, worker threading model
zeus	*Zeus Web Server v4.3* 16K connections, FastCGI

Table 3. DRAM access related parameters for each workload.

Workload	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10
IPC and cache access characterization										
Total IPC over 4 cores	6.07	0.72	3.60	0.44	0.11	2.99	2.93	2.25	2.81	1.16
L2 lookups per 100M cycles [M]	3.58	19.1	17.6	3.82	3.09	11.4	13.8	10.8	9.65	12.0
DRAM accesses per 100M cycles [M]	0.31	0.40	0.36	1.29	1.10	0.88	0.95	1.19	1.18	1.24
L2 miss rate (LRU) [%]	5.47	2.06	1.34	23.4	21.0	4.96	4.23	6.85	7.71	6.56
L2 miss rate (TA-LRU) [%]	5.66	2.26	1.45	23.8	21.8	5.28	4.64	7.89	8.12	7.06
L2 miss rate (CLRU) [%]	5.53	3.39	1.45	24.0	21.9	5.25	4.48	8.18	8.62	7.21
L2 miss rate (TA-CLRU) [%]	5.67	3.92	1.46	24.8	21.7	5.45	4.57	8.54	8.91	7.47
DRAM timing and thermal characterization										
Row buffer miss rate (baseline) [%]	82.6	86.2	79.0	90.2	91.1	85.6	86.4	87.8	88.5	87.3
Row buffer miss rate (TA-MWB) [%]	59.4	84.6	65.8	68.7	90.2	57.2	79.9	71.4	72.6	72.6
Baseline max DRAM temp. [°C]	51.8	52.2	53.1	97.7	93.3	77.4	80.2	92.5	92.0	95.2
Baseline AMB temp. [°C]	71.4	73.2	72.7	112.3	106.1	94.4	97.3	107.8	107.4	109.9



(a) LRU

(b) Clock-based pseudo-LRU

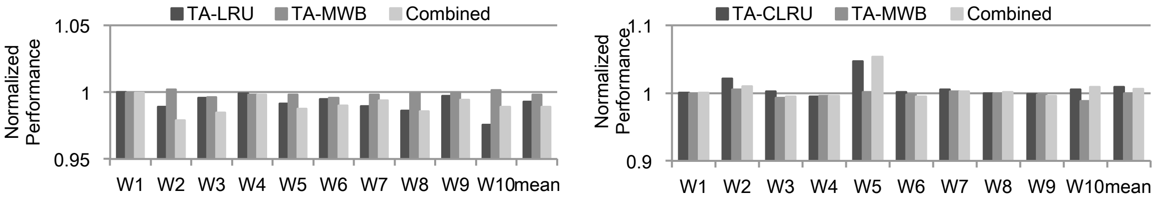
Figure 11. Maximum DRAM temperature reduction in different systems.

maximum DRAM temperature on a DIMM. Therefore, workloads with heavy DRAM traffic will experience fewer thermal emergencies and hence, enjoy performance improvements. In Section 6.3, we show the performance benefits of our techniques on systems that are throttled because of DRAM thermal emergencies.

6.2.1. TA-LRU and TA-MWB

Figure 11 shows the temperature reduction achieved by the TA-LRU scheme, TA-MWB scheme, and a combination of them for the base case of the LRU scheme and CLRU scheme. Most workloads achieve significant maximum DRAM temperature reduction, except for W7 and W8 in the system with the LRU L2 cache, where the maximum temperature increases slightly. The reason is that these two workloads both have heavy L2 cache traffic and a high IPC. Therefore they are very sensitive to different L2 cache replacement policies. Compared with the LRU replacement policy, the TA-LRU scheme may increase the L2 miss rate, and thus increase the DRAM traffic and DRAM temperature. In all other cases, our schemes reduce the DRAM temperature. In addition, our two techniques do not conflict with each other: the combined approach achieves the best temperature reduction. W4 in a system with the combined TA-CLRU and TA-MWB schemes exhibit the highest temperature reduction (8.39°C on the hottest DRAM chip). The average temperature reduction (over all workloads and both baseline systems) achieved by the combined approach is 5.36°C.

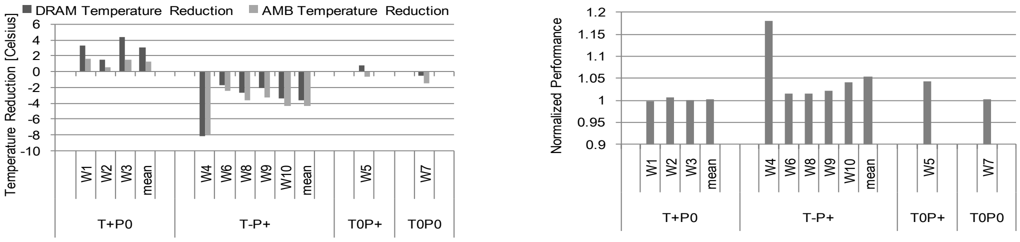
We also evaluate the impact of our techniques on the architectural performance. Figure 12 shows the average instructions per cycle (IPC) for the four cores normalized to the baseline case. On average, our combined temperature aware policies slightly improve the system performance in a system with a clock-based LRU policy, while the performance is reduced by about 1% on average in the systems with a LRU policy. For those applications where our schemes reduce the IPC, the penalty is mainly due to two reasons. First, the presence of the TA-LRU policy may increase the L2 cache miss rate. Second, when an operation in the TA-MWB policy accesses the DRAM,



(a) LRU

(b) Clock-based pseudo-LRU

Figure 12. Normalized performance with respect to the baseline in different systems.



(a) Temperature Reduction

(b) Normalized Total IPC

Figure 13. Temperature reduction and normalized IPC achieved by TA-PA.

it may conflict with a read operation. Maximum performance penalty is observed for W10 on a system with a LRU L2 cache. The TA-CLRU policy increases the L2 cache miss rate by 0.12% (from 3.27% to 3.39%). Since W2 has many L2 cache lookups, the increase in the L2 miss rate hurts the performance. On the other hand, the TA-CLRU policy outperforms the CLRU policy for W5 by over 4.5%. This is because the TA-CLRU policy actually reduces the L2 cache miss rate by 0.2% (from 21.94% to 21.74%). On average, the combination of the TA-LRU and TA-MWB schemes reduce the performance by 1.1% for the LRU policy and increases the performance by 0.6% for the CLRU policy.

6.2.2. TA-PA

Figure 13 shows the temperature reduction and the normalized IPC achieved by our TA-PA scheme on the same ten workloads used in Section 6.2.1. Comparing Figure 13(a) with 13(b), we observe that TA-PA impacts both the temperature and the performance. In order to clearly analyze these impacts, we divide these applications into 4 categories based on their impact on the DRAM operating temperature and performance. Categories are labeled as followed: T *symbol* P *symbol*. T represents temperature, and P represents performance. The symbol following T and P can be +, 0, or -. + indicates an improvement, 0 indicates negligible impact, and – indicates degradation. For example, T+P+ means that the TA-PA policy reduces temperature and improves performance compared to the base case. As shown in Figure 13, W1, W2, and W3 fall into the T+P0 category, with a 3.1°C temperature reduction and a negligible performance penalty. The TA-PA policy achieves a 5.5% average performance benefit on W4, W6, W8, W9, and W10 (T-P+), with a 3.6°C increase in the maximum DRAM temperature. As for W4, the TA-PA policy achieves an 18.2% performance improvement with an 8.2°C increase in the maximum DRAM temperature. W5 exhibits a performance benefit of 4.4% with negligible temperature overhead (T0P+). Finally, W7 incurs a temperature penalty while retaining the same performance (T-P0) because Stage 2 dominates the execution of W7. On average, the TA-PA policy achieves a 3.3% performance improvement for the ten workloads with a 0.8°C increase in the maximum DRAM temperature. The temperature increase does not signify the TA-PA scheme failing to minimize the thermal variation; instead, the overall DRAM temperature increases because of the increased DRAM traffic enabled by the TA-PA scheme.

Figure 14 shows the number of swap operations of a baseline system using clock-based page management [28] and a system with the TA-PA scheme. Compared with the baseline system, the TA-PA scheme reduces the average number of page swaps over these workloads by 7%. This demonstrates that the TA-PA scheme does not incur a performance penalty by increasing the number of page swaps.

To further examine the TA-PA scheme, we performed additional simulations. The relationship between the temperature reduction and the normalized IPC for additional workloads are presented in Figure 15. In these simulations, we run 4 copies of the same application on the simulated CMP. Again, we categorize these applications based on their impacts on the DRAM temperature and performance. As shown in Figure 15, the majority of these applications fall in the first three categories, T+P+, T-P+ and T+P0. These results demonstrate that either (a) the TA-PA scheme effectively reduces maximum DRAM temperature by reducing within DIMM temperature variation or (b) the TA-PA scheme increases the maximum DRAM temperature due to the improved system performance. On average, the TA-PA scheme reduces temperature by 2.2°C and improves performance by 0.9%.

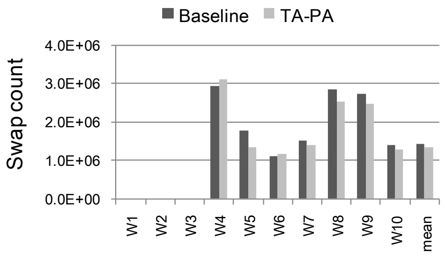


Figure 14. Number of page swaps in baseline system and a system with TA-PA.

Figure 16 shows the temperature reduction achieved by TA-PA on commercial workloads. Note that for these workloads, we simulate a memory with 4 DIMMs. Figure 16 presents both the temperature reductions achieved on each DIMM and the reductions achieved on the maximum temperatures among the 4 DIMMs. For these workloads, the maximum DRAM temperature is reduced by as much as 5.2°C (4.9°C on average). Note that since we simulate these commercial workloads with a large DRAM (16GB), we did not observe any swapping. Various workloads, such as in-memory databases, commonly employ this configuration [29].

Finally, we have also studied a system that combines all of the three proposed optimizations, the TA-LRU scheme, the TA-MWB scheme, and the TA-PA scheme. For the 10 workloads we have studied (Table 2), the combined schemes reduce the DRAM temperature by up to 4.8°C and improves the performance by up to 6.6%. We do not provide the detailed results due to a lack of space. In summary, the combined scheme is less aggressive than TA-PA: the average performance improvement is smaller, but the DRAM temperature reduction is higher.

6.3. Overall System Performance Results

In this section, we take into account DRAM throttling and simulate dynamic thermal management (DTM) for a baseline system and compare it with TA-CLRU and TA-PA systems. We set a temperature threshold of 85°C, which is typical for commercial DRAM devices, to begin throttling of the DRAM to prevent thermal emergencies. Both the baseline system and system with our proposed technique are equipped with the DRAM throttling technique presented in [6] to avoid overheating the DRAM. When evaluating the DRAM transient temperature, we reduce the thermal RC constant to 0.001 times of the original value. In five workloads (W4, W5, W8, W9, W10), one or more DRAM chips exceed their maximum temperature rating (as summarized in Table 3).

Figure 17 shows the normalized total IPC for these workloads when DTM is applied. With the TA-CLRU scheme, the total IPC of W5 is increased by 15.8%. This is because W5 has heavy DRAM traffic, and the TA-CLRU scheme significantly reduces the temperature of W5. On average, the TA-CLRU scheme improves the system performance by 4.1% over these five workloads. Similarly, the TA-PA scheme achieves performance improvement for all these workloads. The maximum performance benefit (9.7%) is observed on W4; while the average performance benefit is 3.6%.

7. Conclusions

In this paper, we leverage upon the fact that thermal variations exist among DRAM chips across a DIMM to prevent thermal emergencies and provide performance benefits. We use experimental data to verify that these thermal variations do exist and to develop a thermal model to estimate the temperature of each DRAM chip. Then, we propose schemes to manage the temperature of the memory modules. Particularly, we propose the Temperature-Aware LRU cache line replacement policy (TA-LRU), the Temperature Aware Memory Write Buffer (TA-MWB) policy, and the Temperature Aware Page Allocation (TA-PA) policy to balance the

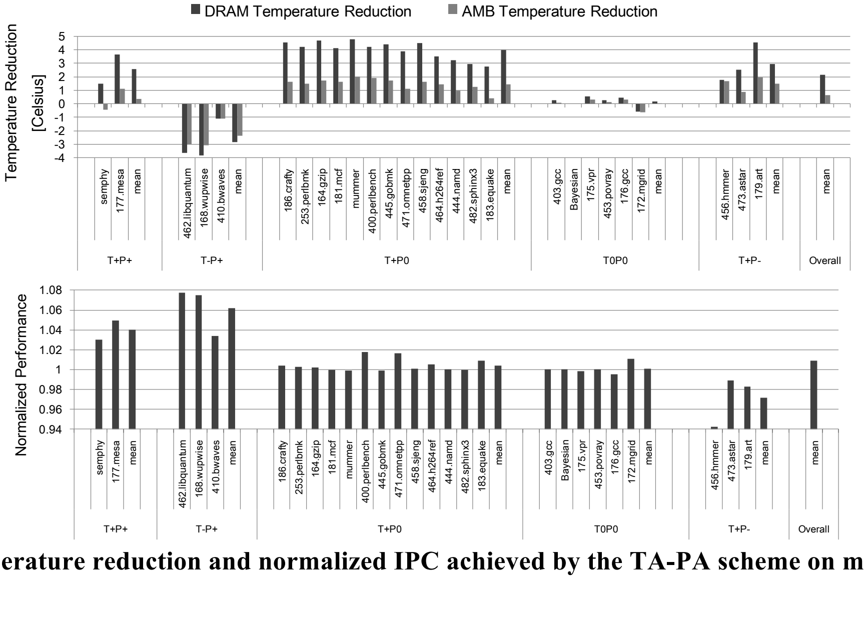


Figure 15. Temperature reduction and normalized IPC achieved by the TA-PA scheme on more workloads.

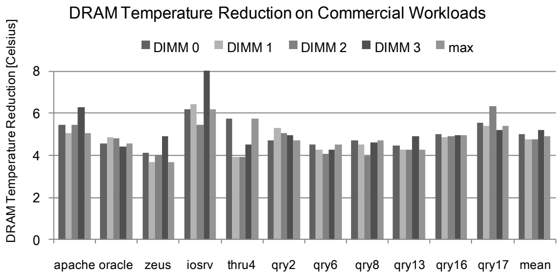


Figure 16. DRAM temperature reduction achieved by TA-PA on commercial workloads.

operating temperatures on the DIMM. Experiments show that a combination of the TA-LRU and TA-MWB policies reduce the peak steady-state temperature of DRAM chips by as much as 8.39°C among 10 workloads and two different baseline systems (5.36°C on average). The TA-PA scheme has a positive impact on DRAM operating temperature as well as system performance on these workloads and also achieves a 4.9°C temperature reduction on 11 commercial workloads. Independently, we show that for workloads that require DRAM Dynamic Thermal Management (DTM), the reduction in temperature achieved by combining the TA-CLRU and TA-MWB policies can result in as much as 15.8% improvement in the system performance (4.1% on average). The maximum and average performance improvements achieved by the TA-PA scheme on these workloads are 9.7% and 3.6%, respectively.

8. Acknowlegements

We would like to thank the anonymous reviewers for their helpful comments. This work is in part supported by NSF Grant CCF-0916746, CCF-0747201 and CNS-0720691.

9. References

[1] Q. Zhu, X. Li, and Y. Wu, "Thermal management of high power memory module for server platforms," in *ITHERM'08.*

[2] J. Dorsey, et al., "An Integrated Quad-Core Opteron™ Processor," IEEE International Solid-State Circuits Conference, 2007.

[3] H. P. Hofstee, "Power Efficient Processor Architecture and The Cell Processor," International Symposium on High-Performance Computer Architecture, 2005.

[4] H. Q. Le, et al., "IBM Power6 Microarchitecture," *IBM Journal on Research and Development,* 51, 2007.

[5] L. Seiler, et al., "Larrabee: a many-core x86 architecture for visual computing," *ACM Transactions on Graphics,* 2008.

[6] J. Iyer, C. L. Hall, J. Shi, and Y. Huang, "System Memory Power and Thermal Management in Platforms Built on Intel® Centrino® Duo Mobile Technology," *Intel Technology Journal,* 2006.

[7] T. Micron, "Technical Note: Uprating Semiconductors for High-Temperature Applications," 2004.

[8] G. E. Suh, S. Devadas, and L. Rudolph, "A New Memory Monitoring Scheme for Memory-Aware Scheduling and Partitioning," in *HPCA8*, 2002.

[9] H. Lee, et al., "Thermal Management of High Power Memory Module " in *SEMI-THERM*, 2006.

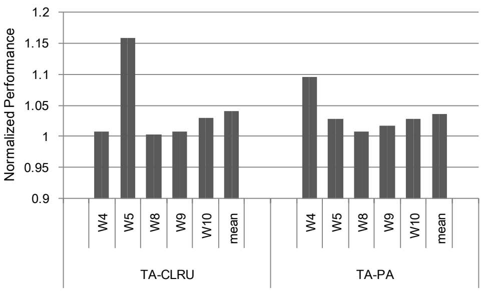


Figure 17. Performance with dynamic thermal management (DTM).

[10] J. Lin, H. Zheng, Z. Zhu, E. Gorbatov, H. David, and Z. Zhang, "Software Thermal Management of DRAM Memory for Multicore Systems," SIGMETRICS'08, 2008.

[11] V. Delaluz, et al., "Hardware and Software Techniques for Controlling DRAM Power Modes," *IEEE Transactions on Computers,* 50, 2001.

[12] X. Fan, C. S. Ellis, and A. R. Lebeck, "Memory Control Policies for DRAM Power Management," ISLPED'01, 2001.

[13] H. Huang, P. Pillai, and K. G. Shin, "Design and Implementation of Power-Aware Virtual Memory," the USENIX Annual Technical Conference, 2003.

[14] A. R. Lebeck, X. Fan, H. Zeng, and C. Ellis, "Power Aware Page Allocation," presented at the ASPLOS-IX, 2000.

[15] J. Lin, H. Zheng, Z. Zhu, H. David, and Z. Zhang, "Thermal Modeling and Management of DRAM Memory Systems," presented at the ISCA'07, 2007.

[16] S. Liu, S. Ogrenci Memik, Y. Zhang, and G. Memik, "A Power and Temperature Aware DRAM Architecture," in *DAC'08*, 2008.

[17] S. Liu, S. Ogrenci Memik, Y. Zhang, and G. Memik, "An Approach for Adaptive DRAM Temperature and Power Management," presented at the ICS'08, 2008.

[18] K. Skadron, T. Abdelzaher, and M. R. Stan, "Control-theoretic techniques and thermal-RC modeling for accurate and localized dynamic thermal management.," in *HPCA'02.*

[19] G. E. Suh, L. Rudolph, and S. Devadas, "Dynamic Cache Partitioning for Simultaneous Multithreading Systems," in *ICPDCS'01*, 2001.

[20] H. Dybdahl, P. Stenstrom, and L. Natvig, "A Cache-Partition Aware Replacement Policy for Chip Multiprocessors " presented at the ACM 2006 Conference on High Performance Computing 2006.

[21] H. Al-Zoubi, A. Milenkovic, and M. Milenkovic, "Performance Evaluation of Cache Replacement Policies for the SPEC CPU2000 Benchmark Suite," ACMSE'04, 2004.

[22] Micron, "System Power Calculator,

[23] G. H. Loh, S. Subramaniam, and Y. Xie, "Zesto: A Cycle-Level Simulator for Highly Detailed Microarchitecture Exploration," presented at the ISPASS'09, 2009.

[24] www.spec.org, "Standard Performance Evaluation Corporation. SPEC CPU2000. ,"

[25] R. Narayanan, B. Ozisikyilmaz, J. Zambreno, J. Pisharath, G. Memik, and A. Choudhary., "MineBench: A Benchmark Suite for Data Mining Workloads," in *IISWC'06.*

[26] K. Albayraktaroglu, et al., "BioBench: A benchmark suite of bioinformatics applications," ISPASS'05.

[27] F. W. Thomas, E. W. Roland, F. Michael, A. Anastassia, F. Babak, and C. H. James, "SimFlex: Statistical Sampling of Computer System Simulation," IEEE Computer Society Press, 2006.

[28] F. J. Corbato, "A Paging Experiment with the Multics System," *MIT Project MAC Report MAC-M-384,* 1968.

[29] J. Belzer, *Encyclopedia of Computer Science and Technology - Volume 14: Very Large Data Base Systems to Zero-Memory and Markov Information Source*: Marcel Dekker Inc.