An Approach for Adaptive DRAM Temperature and Power Management

Song Liu, Seda Ogrenci Memik, Yu Zhang, and Gokhan Memik
Department of Electrical Engineering and Computer Science
Northwestern University, Evanston, IL 60208, USA
{sli646, seda, yzh702, memik}@eecs.northwestern.edu

ABSTRACT

With rising capacities and higher accessing frequencies, high-performance DRAMs are providing increasing memory access bandwidth to the processors. However, the increasing DRAM performance comes with the price of higher power consumption and temperature in DRAM chips. Traditional low power approaches for DRAM systems focus on utilizing low power modes, which is not always suitable for high performance systems. Existing DRAM temperature management techniques, on the other hand, utilize generic temperature management methods inherited from those applied on processor cores. These methods reduce DRAM temperature by controlling the number of DRAM accesses, similar to throttling the processor core, which incurs significant performance penalty. In this paper, we propose a customized low power technique for high performance DRAM systems, namely the Page Hit Aware Write Buffer (PHA-WB). The PHA-WB improves DRAM page hit rate by buffering write operations that may incur page misses. This approach reduces DRAM system power consumption and temperature without any performance penalty. Our proposed Throughput-Aware PHA-WB (TAP) dynamically configures the write buffer for different applications and workloads, thus achieves the best trade off between DRAM power reduction and buffer power overhead. Our experiments show that a system with TAP could reduce the total DRAM power consumption by up to 18.36% (8.64% on average). The steady-state temperature can be reduced by as much as 5.10°C and by 1.93°C on average across eight representative workloads.

Categories and Subject Descriptors

B.3.2 [**Memory Structure**]: Design Styles – Primary Memory

General Terms

Management, Design, Performance

Keywords

DRAM, Power, Temperature

ICS'08, June 7–12, 2008, Island of Kos, Aegean Sea, Greece.

1. INTRODUCTION

Technological advances in microprocessor architectures enable high performance with an underlying assumption on increasing utilization of the memory systems. Particularly, Chip Multi Processors (CMPs) led to significant rises in computation capacity and subsequently to the need for larger data bandwidths to be supported by DRAM systems. Fully Buffered Dual In-line Memory Module (FB-DIMM) architecture [5] provides increasing main memory capacity. On the other hand, while the new DRAM architectures are aiming at responding to the increasing data bandwidth needs, increasing memory densities and data rates lead to higher operating temperatures in DRAM systems. Prior studies have shown that DRAM temperature control has become a practical and pressing issue as a result of these trends [4]. Experiments on several thin-and-light laptops show that 1GB SO-DIMMs reach temperatures of up to 85°C, exceeding their maximum case temperature [4].

A recent study investigated DRAM performance under thermal and power constraints [4]. Mechanisms such as thermal sensors and Delta Temperature (DT) in Serial Presence Detect (SPD) have been used to predict DRAM temperature and apply thermal management when necessary. Lin et al. [9] proposed an adaptive core gating and dynamic voltage and frequency scaling (DVFS) approach in CMP systems to control DRAM temperature. Both solutions are generic thermal management techniques developed primarily for processor core temperature control. These methods achieve thermal control by trading off performance.

In order to develop more efficient solutions, we need to understand the inherent operating principles of DRAM structures and develop schemes to reduce the thermal implications. In this paper, we develop such a scheme. Particularly, we propose a customized solution, which specifically aims to address the power and thermal behavior of DRAMs. Our goal is to provide a solution that can harvest the largest peak temperature reduction without incurring any performance overhead. Specifically, we propose a customized method by improving DRAM page hit rate. Our technique does not impose any limitations on the co-existence of other complementary techniques that

trade-off performance if more aggressive thermal management is necessary.

In modern DRAM systems, I/O modules cannot access DRAM arrays directly. The target row (or page) should be opened (activated) before it can be read or written. Page hit rate is defined as the percentage of read/write operations that target an already opened page. When there is a page miss, the system has to spend time and power to activate the new target row. Therefore, improving page hit rate could in fact improve DRAM performance while reducing DRAM power consumption and temperature.

From an architectural point of view, DRAM read operations are the bottleneck of system performance, while write operations are not the primary concern. We exploit this fact through an enhancement to the DRAM architecture. We introduce a buffering mechanism to hold write operations that may cause a page miss. Specifically, we propose to embed a Page Hit Aware Write Buffer (PHA-WB) mechanism into the DRAM to utilize the relationship between page hit rate and power consumption. The buffered writes are placed into the DRAM later when their target row is activated by read operations. As a result, the write buffer improves page hit rate significantly, and thus reduces DRAM power consumption and temperature.

We also aim our temperature-aware scheme to be able to dynamically adjust the trade-off between the aggressiveness of the power optimization mechanism and resulting temperature reduction at the expense of more storage for buffering the data and orchestrating the buffer-DRAM coordination. Specifically, throughout the execution of an application, power consumption will vary depending on the nature of the application and the workload, resulting in varying power densities and operating temperatures. A large write buffer may yield more DRAM power saving and temperature reduction at the expense of more complex circuitry for buffering the data and comparing target addresses. Therefore, there is a need to effectively adjust the aggressiveness of our approach to find the optimal operating conditions. In order to achieve this, we propose an Throughput-Aware PHA-WB (TAP), which considers different DRAM access patterns and dynamically chooses the best configuration.

Our experiments show that our proposed TAP reduces the total DRAM system power consumption by as much as 18.36% (8.64% on average) and DRAM steady-state temperature by as much as 5.10°C (1.93°C on average) for eight different workloads based on 20 SPEC CPU 2000 benchmarks running on a 4-core CMP [18].

The remainder of the paper is organized as follows. Section 2 discusses related work. Section 3 provides an overview of modern DRAM systems. Our proposed technique is described in Section 4. Section 5 presents experimental methodology and results. Finally, we conclude the paper in Section 6.

2. RELATED WORK

DRAM power consumption can comprise a large portion of the total system power, particularly for mobile and embedded systems. In order to save energy consumed during idle period between DRAM access, modern DRAMs provide various power-saving modes. One example is Rambus DRAM (RDRAM), which has 4 power modes: active, standby, nap, and power down [15]. Various low power DRAM techniques focus on utilizing these idle modes efficiently to achieve best energy-delay product [2, 3, 7]. Our goal however, is to tie the active periods of DRAM operation to power consumption and thermal behavior. Hence, these *power mode based* techniques can be considered as complementary to our proposed approach.

Reorder memory controller is a widely used technique in stream processors [8, 16]. In these systems, the memory controller reorders memory accesses, so that there are more chances to use efficient page mode and burst mode. On the other hand, our technique targets general propose processors with write back cache and FB-DIMM, where each memory command access one cache line with burst mode. Since burst mode access is already a highly "in-order" memory access, memory reordering could not yield significant improvement. On the other hand, our approach is a further enhancement for burst-accessed DRAM. The TAP provides flexible reordering as well as data buffering.

DRAM thermal management has become a pressing issue in mobile systems [4]. On-DIMM thermal sensors or delta temperature (DT) in Serial Presence Detect (SPD) are used to detect overheated DRAM chips in mobile systems [2]. Dynamic thermal management (DTM) techniques such as thermal shut down and DRAM bandwidth throttling is used to control the DRAM activity, hence, temperature. In order to cool down the DRAM while keeping the performance penalty small, Lin et al. [9] proposed adaptive core gating and dynamic voltage and frequency scaling (DVFS) to CMP systems aiming to achieve the maximum performance with DRAM thermal constraints. DTM and DVFS are generic thermal management techniques that have been applied for processors and disk drives. However, they are known to introduce system performance penalties, because their end effect is a direct reduction of DRAM accesses available per unit time. We refer to these techniques as *memory traffic control based* temperature aware techniques, since they handle DRAM thermal emergencies by reducing DRAM access density.

Existing power and temperature aware techniques focus on two special cases. The power mode based techniques are designed for applications with fewer DRAM accesses. Therefore, the key idea is to maximize the use of low power

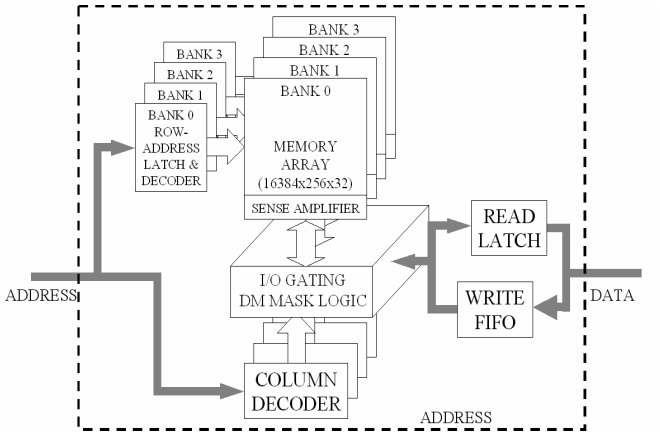

Figure 1. System diagram of DDR2 SDRAM

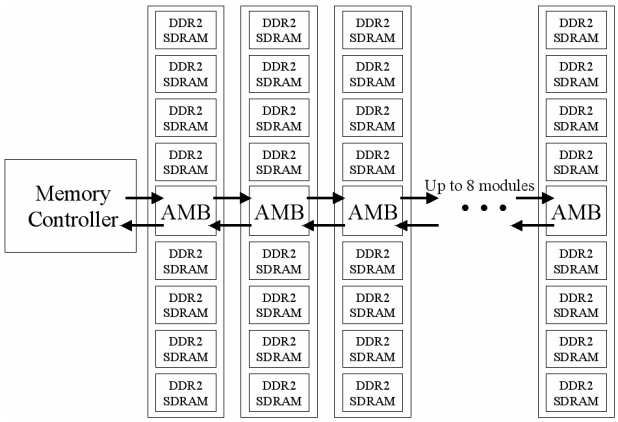

Figure 2. System diagram of FB-DIMM

modes, with low performance penalty. On the other hand, memory traffic control based techniques are applicable when the DRAM traffic is heavy. Our proposed technique is designed for the later case. However, instead of applying thermal management as an intervention mechanism after the DRAM is overheated, our technique prevents potential thermal emergencies by improving page hit rate and thus reducing the power consumption. This is achieved by accessing the DRAM in an intelligent way. Since our technique does not use low power mode or throttle the DRAM bandwidth, it does not incur performance penalties (in fact increasing DRAM page hit rate may improve the system performance). Moreover, our technique could easily be integrated with memory traffic control based techniques for more aggressive thermal management.

To our best knowledge, the power optimization technique by Lin et al. [10] could be considered as the closest related work. They propose an optimization, which also addresses the active power consumption. They introduce a small prefetching cache in the DRAM to improve system performance and reduce power consumption. However, they use a specific assumption on the baseline system (operation in close page mode), where the page hit rate is forced to be zero. However, in a high performance DRAM system, an alternative configuration (open page mode) is a more likely choice, which draws on the principle of data locality. In open page mode the power and temperature benefit of prefetching would be minimal if not non-existent. Our technique on the other hand, is devised upon and evaluated with a baseline system operating in the high performance open page mode.

3. BACKGROUND

In modern computers, synchronous dynamic random access memory (SDRAM) has replaced other types of DRAM due to its greater speed. Fully Buffered Dual In-line Memory Module (FB-DIMM) and Double Data Rate two (DDR2) SDRAM increase the DRAM bandwidth to keep pace with the increasing memory bandwidth demands by processors. In order to explore potential power and temperature optimizations in DRAM system, it is necessary to understand the DRAM operation mechanism. In this section, we briefly review modern DRAM technology.

3.1 DDR2 SDRAM

Figure 1 illustrates a simplified system diagram based on the DDR2 SDRAM, Micron Technology Inc. [13]. The memory array is hierarchically organized as banks, rows, and columns. In this example, the DRAM chip is divided into 4 banks. Column is the minimum access unit of the SDRAM.

DDR2 SDRAM has five kinds of commands: READ, WRITE, ACTIVATE, PRECHARGE, and REFRESH. I/O modules in SDRAM cannot access the DRAM array directly. Instead, before any READ or WRITE can be issued, a row in that bank must be loaded to the I/O buffer, namely opened or activated, by an ACTIVATE command. Only one row of a bank can be opened at the same time, therefore an opened row must be written back to the memory array (closed) by a PRECHARGE command before another row in the same bank is accessed. DRAM rows have to be refreshed periodically by REFRESH command and when other parts of the system are power down, the DRAM could maintain data by an automatic SELF REFRESH command.

The I/O buffer can be managed in two modes: the open page mode and the close page mode. When the SDRAM is in open page mode, an opened (activated) row is kept open for as long as possible, until this row is written back to the memory array by a REFRESH command or another row in the same bank is accessed. On the other hand, when the SDRAM is working in close page mode, any opened row is precharged immediately after an access. Due to the locality in memory accesses open page mode could provide better performance for most applications.

3.2 FB-DIMM (Fully Buffered DIMM)

Figure 2 illustrates the architecture of FB-DIMM [12]. FB-DIMM introduces an advanced memory buffer (AMB) between the memory controller and DDR2 SDRAM chips. In FB-DIMM system, the memory controller reads or writes through a serial interface to the AMB instead of the parallel-organized SDRAM chips. FB-DIMM provides necessary DRAM system parameters in serial presence detect (SPD), while keeping the organization of multiple DRAM chips transparent to the memory controller. Therefore AMB is the key point of increasing performance, portability, and reliability of FB-DIMM. AMB also offers error detection and correction, without posing overhead on the processor or the memory controller. However, the AMB unit has high power consumption, which makes the FB-DIMM more susceptible to thermal emergencies.

4. ADAPTIVE MANAGEMENT FOR POWER AND TEMPERATURE

This section discusses our propose technique for adaptive DRAM temperature and power management. We first explore potential savings in DRAM power consumption. Then, we describe our Page Hit Aware Write Buffer (PHA-WB) and analyze its overhead. We also introduce an adaptive enhancement, namely the Throughput-Aware PHA-WB (TAP). TAP could achieve optimal tradeoff in different scenarios. Finally, we discuss the implementation of our technique.

4.1 DRAM Power Modeling

DDR2 DRAM power consumption has three main components: background power, activate power, and read/write power [11]. Other power components, such as refresh power, are negligible compared to these components. Background power is power consumption without any DRAM accesses. Different power modes have different background power consumptions. Power mode based techniques focus on reducing background power consumption. Activate power is the power consumption of loading one row to the I/O buffer. Read/write power is the power consumption of reading or writing data stored in the I/O buffer. Memory traffic control based techniques aim to reduce both activate power and read/write power. However, reduction in read/write power occurs at the expense of performance. Therefore, in order to reduce power consumption with no performance penalty, we focus on reducing the activate power.

In close page mode, activate power is proportional to DRAM access frequency. In open page mode, activate power is a function of page hit rate. In other words, higher page hit rate means same number of read and write operations could be completed with fewer activate operations. Therefore, improving page hit rate could achieve power benefit without hurting performance. Therefore, we turn our attention towards opportunities to improve the page hit rate.

4.2 PHA-WB: Page Hit Aware Write Buffer

DRAM read operations pose the major bottleneck of system performance. When reading, the processor has to wait for the data before executing other instructions. Although instruction/task parallelism techniques can be used to hide a portion of this latency, it is known that for many application domains, the memory read latency is one of the most important determinants of performance. On the other hand, DRAM write operations could possibly be delayed without hurting performance. Therefore, modern processors often write to the main memory through a write buffer. In our scheme, we utilize a similar notion. Specifically, we introduce a Page Hit Aware Write Buffer (PHA-WB) for DRAM page hit rate optimization.

PHA-WB is a buffer placed between the memory controller and the DRAM chips. PHA-WB can be implemented both on the memory controller or the DRAM itself; in fact, we propose to implement the PHA-WB in the AMB of FB-DIMM. PHA-WB is transparent to read operations. Read operations pass through the buffer without delay. On the other hand, write operations that are not targeting an activated row will be buffered to achieve higher hit rate. The PHA-WB checks the target address of each operation and maintains a table of the activated rows in each bank. When an operation accesses the DRAM, the target address is broadcasted in the PHA-WB. Buffered write operations with matching rows will access the DRAM after this operation and thus improve DRAM page hit rate. In other words, the main principle used in PHA-WB is to buffer the write operations until a matching read operation is posted. Once such a read operation is initiated (and hence the corresponding row is opened), the corresponding write operation will be completed following the read operation. In the next section (Section 4.3), we describe in detail the hardware structures implemented to perform these operations.

Overall, the PHA-WB observes the following procedures for governing the DRAM accesses in different conditions:

1. Write operations that write to an opened row are sent directly to the DRAM.
2. Write operations that write to a closed row will be buffered in the PHA-WB.
3. Read operations that do not read the data residing in PHA-WB are sent directly to the DRAM, regardless of whether the row is activated or not.
4. Read operations that require the data residing in the PHA-WB are also sent to the DRAM. However, the corresponding data read from the

DRAM will be updated by the data from PHA-WB in the read FIFO.

Write operations buffered in the PHA-WB will be written to the DRAM in the following cases:

1. When the PHA-WB is currently full and another write operation needs buffering. Then, the oldest write operation in the PHA-WB will access the DRAM.
2. When the target row of a write operation is opened by another operation (*row match*), the write operation will access the DRAM after that operation.
3. When a read operation accesses the data residing in the PHA-WB (*address match*), the write operation will access the DRAM after that operation. In the meanwhile, the corresponding data in the read FIFO is replaced with the corresponding data from the PHA-WB. In other words, the read data is provided from the PHA-WB instead of the memory (much like forwarding in a load-store buffer).

When the PHA-WB has a large number of entries, selecting the oldest write operation may be expensive. In this case, the system could randomly choose a buffer entry to be vacated. Our experiments revealed that this will not degrade the optimization quality significantly. The page hit rate varies only by 1.37% on average between choosing the oldest versus choosing a random entry from a PHA-WB with 64 entries. Therefore, our experimental framework uses a random replacement strategy.

Note that for the FB-DIMM, the data read from the DRAM is synchronized in the read FIFO. Therefore our technique will not incur extra delay for read operations.

Another important aspect of the PHA-WB is that it can work with DMA engines. Specifically, a write operation to the memory occurs when a) the processor writes a value or b) the Direct Memory Access (DMA) engine writes data read from an input/output device. In either case, the value that is written is not needed until a read operation to it occurs. Our PHA-WB will capture such cases and guarantee correct operation without any performance penalty.

Note that although PHA-WB only buffers write operations, it also improves page hit rate of read operations, and thus reduces average read delay. Consider the following example illustrating such cases. Assume that the processor reads an array while writing to another one. If these two arrays are stored in two different rows in the DRAM, the DRAM has to switch between these two rows and suffers from a large number of page misses. With the PHA-WB, these write operations are buffered. The DRAM could keep one row open until all the read operations are finished, and the write operations are buffered and they will access the DRAM later. Therefore, PHA-WB could improve both read and write page hit rate, and thus, reduce average read delay.

4.3 Throughput-Aware PHA-WB

The number of entries in the PHA-WB is a tradeoff between buffer power overhead and DRAM power savings. More entries will yield more DRAM power savings and temperature reduction at the expense of more storage for buffering the data and comparing target addresses (and hence increased power consumption). Considering each write operation will write 64 bytes to the DRAM, we compare different configurations of PHA-WB with 16, 32, and 64 entries, which correspond to 1KB, 2KB, and 4KB memory added to the AMB. A detailed comparison of these configurations is presented in Section 5.

In order to achieve better power savings and more aggressive control over the power density of the DRAM, we introduce adaptive adjustment to PHA-WB, which could dynamically choose the size of the write buffer according to different DRAM access patterns. Experiments show that increasing the size of the write buffer could generally increase the page hit rate consistently. However, increasing the page hit rate may or may not lead to significant reduction in power consumption. The DRAM power consumption is a function of both page hit rate and DRAM throughput. Applications with high DRAM throughput could benefit more from an improvement in DRAM page hit rate. Therefore, we could dynamically determine the size of the write buffer by monitoring the DRAM throughput within a given period of time and enable an *Throughput-Aware* PHA-WB (TAP).

The TAP employs a performance counter. Particularly we implement a counter in TAP, which tracks the number of clock cycles for each k consecutive DRAM accesses. Then, the write buffer size is set according to this rate. More cycles for k consecutive DRAM accesses means the DRAM throughput is low at that time. Therefore, we could determine the size of the write buffer by checking the counter. Moreover, when the processor switches between two different applications or two distinct phases of an application, the size of the buffer could also be switched. TAP utilizes clock gating to modify the number of activated entries in the write buffer. When the write buffer size is reduced, buffered data is written into memory when the DRAM is not busy, regardless of whether this will incur a page miss or not.

Since such release operations will happen when the DRAM throughput is low, buffer down-sizing will not take an excessively long time. However, frequent switches will diminish the power savings achieved through the TAP. In order to minimize the number of switches, our technique tracks several consecutive samples from the performance

counter and makes the switch if and only if all these results suggest a re-sizing. Our experiments show that by tracking 4 consecutive intervals, with each interval monitoring the total duration to finish 10 DRAM accesses, the number of switches is less than 10 times per second. Our experimental results also reveal that this rate achieves optimal results for the TAP.

4.4 Implementation of the TAP

Since the TAP targets high performance DRAM systems. We further discuss how to implement the TAP in FB-DIMM. In FB-DIMM, the best choice is to implement the TAP in the AMB [6]. DRAM commands are decoded in the AMB, so that the target address could be analyzed in the AMB. In current AMB, a deep write buffer is employed to store the data exchanges between the memory controller and the DDR I/O port. We replace this buffer with the TAP, so that the structure of the AMB does not change. Another benefit of implementing the TAP in AMB is that it is fully associated with all DRAM banks. Therefore the buffer entries are utilized efficiently. Finally, since some read operations may access data residing in the buffer, TAP needs to search the buffer for corresponding data. Since it takes much longer time for the AMB to access DRAM than to search the buffer, implementing TAP in the AMB will not introduce extra delay to read operations in any scenario.

Figure 3 illustrates the structure of the TAP implemented in the AMB for a DRAM system with *m* independent DRAM banks, and *n* TAP entries. The TAP is enclosed in the dashed rectangle. TAP has three main parts: the Activated Rows Table, the Write Buffer, and the Adaptive Adjustor. Activated Rows Table is a table with *m* entries. Each entry traces the activated row of an independent bank. The Write Buffer is composed of a content addressable memory (CAM), which keeps write addresses, and a buffer array, which keeps the write data. The Adaptive Adjustor counts number of DRAM accesses in units time. When the DRAM traffic condition is changing, the Adaptive Adjustor adjusts the number of activated entries correspondingly.

The TAP works as follows. DRAM operations are decoded by the command decoder. Read operations are not subject to buffering, therefore, read operations are sent to the operation queue directly. On the other hand, for all write operations, the Activated Rows Table must be checked for a possible match between their target row and the row that is already activated. If the target row is currently activated, the write operation is sent to the operation queue, if not, the operation is placed in the Write Buffer. For each operation that has just entered the operation queue, its target address is sent to the Activated Rows Table for an update. In the meanwhile, the address is broadcasted in the Write Buffer. Each entry compares the broadcast address with its own target address. If these two addresses are identical, we refer to this case as an *address match*. If the two addresses are

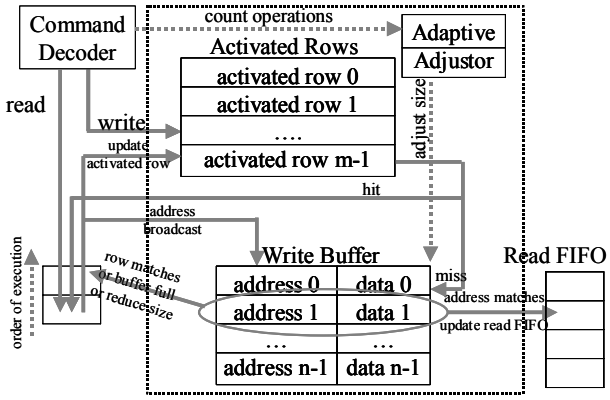

Figure 3. Block diagram of the PHA-WB scheme

Table 1. Power Consumption of PHA-WB with different sizes under 90 nm Technology

PHA-WB Size	PHA-WB Power Consumption (W)
8-entry	0.093970
16-entry	0.098326
32-entry	0.107038
64-entry	0.127537
128-entry	0.222912

targeting the same row, we call it a *row match*. Address match is a special case of row match. Write operations with row matches are sent to the DRAM after the current operation, because this operation will hit an activated row. Write operations with address matches are also sent to the Read FIFO and replace the data read by the current read operation, because the data read from the DRAM is outdated. Due to limited size, it is necessary to issue buffered write operations regardless hit or not. When the Write Buffer is full and a new write operation does not hit an activated row, a randomly selected operation in the TAP is sent to the operation queue and the new write operation enters the TAP. Finally, when the Adaptive Adjustor reduces the number of activated entries. Write operations residing in the entries to be clock gated are issued to the DRAM.

The Adaptive Adjustor counts the number clock cycles used for consecutive *k* DRAM operations. The size of the Write Buffer is adjusted correspondingly.

We have estimated the power consumption of PHA-WB on a 1GB FB-DIMM system by modeling the broadcast structure and the buffer array using CACTI 3.2 [17]. The power overheads with different PHA-WB sizes are given in Table 1. The power consumption of PHA-WB is small compared to the DRAM power consumption, which is over 3W on average among our experimental workloads. We observe from the table that the power overhead does not increase significantly for up to the 64-entry design. However, 128-entry PHA-WB is much more power

Table 2. Experimental configuration of processor, memory system, and PHA-WB.

Parameters	Values
Processor	4-core, 2.5GHz
Instruction Cache (per core)	64KB, 2-way, 64B line, 1 cycle hit latency, 16 MSHRs
Data Cache (per core)	64KB, 2-way, 64B line, 2 cycle hit latency, write-back, 16 MSHRs
L2 Cache (shared)	16MB, 8-way, 128B line, 13 cycle hit latency, write-through, 40 MSHRs
FB-DIMM	1GB, 512Mb per chip, 8 DQs per chip, 667MHz, 250 cycle baseline delay
DDR2 DRAM Chip	4 banks per chip, 16384 rows per bank, 256 columns per row, 4 Bytes per column
Burst Mode	Burst length of 8
Major DRAM Timing Parameter	Active to Read tRCD=15ns, Read to Read Data tCL=15ns, Precharge to Active tRP=15ns
No. of 64 Byte Entries in PHA-WB	16/32/64/*adaptive*

consuming than smaller designs. Therefore, the number of entries in our TAP switches among 16, 32, and 64.

5. EXPERIMENTAL RESULTS

In this section, we first introduce the power and thermal models used in our evaluation. Then, we outline our experimental methodology and describe our benchmarks. Finally, we present the results demonstrating the effectiveness of our proposed technique.

5.1 Power and Thermal Modeling

We use a DRAM power model based on the data published by Micron Technology Inc. for DDR, DDR2, and DDR3 memory chips [14]. We utilize an RC-analogy based thermal model widely used in literature for the DRAM system, which is also similar to the model presented by Lin et al. [9]. The DRAM system is modeled at chip level as the DRAM chips and the AMB. The thermal resistance from the AMB and the DRAM chips to the ambient are 9.3°C/W and 4.0°C/W, respectively. The thermal resistance from the DRAM chips to the AMB is 3.4°C/W and the thermal resistance from AMB to DRAM is 4.1°C/W. We present static temperature values for each workload. We also adopted the AMB power model used by Lin et al. [9]. We have estimated the power consumption of PHA-WB by modeling the broadcast structure and the buffer array using CACTI 3.2 [17]. This power overhead of the PHA-WB has been considered in our evaluation.

5.2 Experimental System Configuration

We used M5 simulator [1] as our architectural simulator to extract memory read and write traces for SPEC CPU 2000 applications. We assume a CMP with 4 processors and 1 GB DDR2 FB-DIMM. Each processor executes one of the SPEC applications (the workloads are described in Section 5.3). The major parameters for the processor and memory are listed in Table 2. Then, we analyze the DRAM page hit rate under different configurations with and without PHA-WB. System Power Calculator [14] is used to calculate DRAM power consumption. Finally, we calculate AMB power and static DRAM system temperature for each trace based on these calculations.

Table 3. Workload Mixes

Workload	Benchmarks
W1	swim, swim, swim, swim
W2	swim, lucas, applu, mgrid
W3	wupwise, apsi, fma3d, facerec
W4	swim, lucas, wupwise, apsi
W5	swim, lucas, sixtrack, galgel
W6	equake, gap, gcc, wupwise
W7	ammp, apsi, vpr, parser
W8	mcf, vortex, mesa, gzip

In order to evaluate the page hit rate, we define the memory mapping as follows. Read and write accesses are burst-oriented, in DDR2 SDRAM, with the burst length being programmable to either 4 or 8. The burst length determines the maximum number of column locations that can be accessed for a single read or write command. We chose burst length of 8 in our simulations. For 1GB DRAM module, 30 bits (bit 0 to 29) are used to describe the memory space. Bit 1 and 0 are mapped to each byte within a column. Bits 4 through 2 are mapped to adjacent columns that could be burst accessed from the DRAM. Therefore, burst read and write could access 32 bytes from one bank. Note that, the size of the L2 cache line is 128 bytes, so each read operation reads 128 bytes from the DRAM. The write mechanism of L2 cache is write- through, so each write operation writes 64 bytes to the DRAM. Therefore, each time at least 64 bytes are accessed at the same time. We map bit 5 to different banks, so that 64 bytes could be accessed with one burst operation. Bit 10 through 6 are mapped to different sections of a row. There are 64 banks in the DRAM module and we have grouped them into groups of two, so there are 32 groups. Bits 15 through 11 are mapped to these groups. Finally, bits 29 through 12 are mapped to different rows within a bank.

We evaluate PHA-WB with 16, 32, and 64 entries as well as the TAP. In TAP, the Adaptive Adjustor monitors the total clock cycles for 10 consecutive DRAM accesses. If 10 DRAM accesses take less than 500 cycles, 64 buffer entries

are enabled; if 10 accesses take between 500 and 1000 cycles, 32 buffer entries are enabled; if 10 accesses take between 1000 and 4000 cycles, 16 buffer entries are enabled; if 10 accesses take more than 4000 cycles, all the buffer entries are disabled and the system works as if there is no write buffer. Switching between different sizes of the write buffer occurs when 4 consecutive counter samples suggest the need for re-sizing.

5.3 Workloads

We used 20 applications from SPEC CPU 2000 benchmarks. In order to represent different DRAM usage, we organized 8 workloads using these applications. Each of these workloads has 4 applications, which are individually executed on a single core. W1 has four instants of swim, which represents the application with highest DRAM accesses. W2 through W8 are various combinations of SPEC CPU 2000 benchmarks. W2, W3, and W4 represent applications with a high number of memory accesses. W7 and W8 represent applications with infrequent memory accesses. W5 and W6 represent cases between these two extremes. The workload contents are described in Table 3.

5.4 Results

We evaluate the impact of the PHA-WB on page hit rate, power consumption, and the static temperature of the DRAM. We performed a comparison of PHA-WB structures with different number of entries. Figure 4 shows the page hit rate of different workloads on systems without PHA-WB, 16-entry PHA-WB, 32-entry PHA-WB, 64-entry PHA-WB, and the TAP. We observe that the page hit rate increases by up to 19.64% (14.45% on average) comparing an architecture without PHA-WB and one with a 16-entry PHA-WB. The maximum and average page hit rate improvement for the 64-entry PHA-WB is 23.32% and 16.43%, respectively. The introduction of even a small sized PHA-WB makes a significant impact compared to the base case. As we increase the PHA-WB size further, we observe a gain; however, the benefits diminish because most of the writes that access a later read row are captured even with a smaller PHA-WB.

Figure 5 illustrates the percentage power savings under different PHA-WB sizes, relative to a system without PHA-WB. The power saving is a function of page hit rate and memory throughput. Some applications with high improvements in page hit rate fail to yield significant power savings because the DRAM throughput is very low. The power savings of the 16-entry PHA-WB design could be as high as 15.57% (7.21% on average). The average and peak page power savings for the TAP are 18.36% and 8.64%, respectively. The TAP yields slightly better results for workloads W1, W3, W4, W6, and W7 compared with the best fixed size PHA-WB. The reason for this improvement is that the TAP could switch during different phases of the

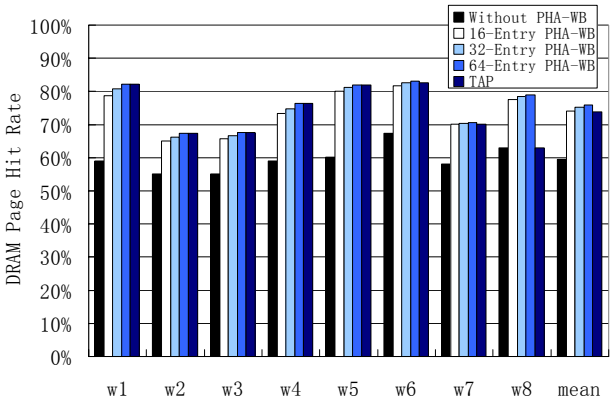

Figure 4. DRAM page hit rate for different workloads with different PHA-WB

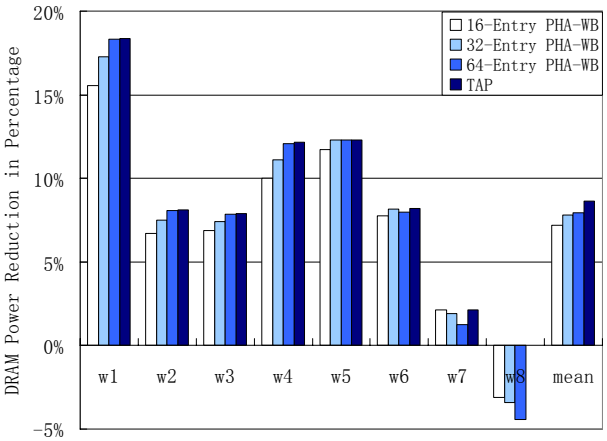

Figure 5. DRAM power reduction for different workloads with different PHA-WB

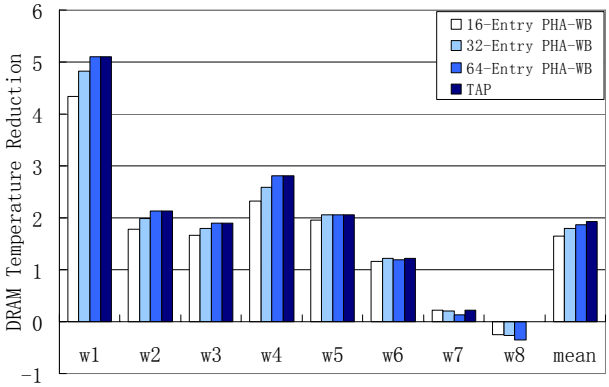

Figure 6. DRAM temperature reduction for different workloads with different PHA-WB

application, thus get better total power reduction. The TAP is entirely disabled when running workload W8, hence, the total power consumption for our architecture equals the base case. Similarly, the steady-state temperature, read delay, and IPC (presented in Figure 6, 7, and 8) are identical to the base case. Fixed size PHA-WBs are not disabled when running workload W8, thus yield negative power and temperature reduction (i.e. degradation in power and temperature).

Figure 6 shows the reduction in DRAM temperature using different PHA-WB sizes compared to the base case without the PHA-WB. The temperature is reduced by up to 5.10°C (1.93°C on average over all workloads) for the TAP and up to 4.30°C (1.65°C on average) for the 16-entry PHA-WB.

We observe that applications with high DRAM throughput accesses are likely to benefit from the PHA-WB. On the other hand, applications with low DRAM throughput do not benefit from the PHA-WB: when there are few DRAM accesses, the background power dominates the total power consumption and the addition of the PHA-WB may in fact increase the overall power consumption as we have discussed previously. In the most general case, a combination of a power mode based low power technique and our approach will succeed in reducing power consumption in both cases. In fact, we see that the TAP always provides a reduction in power or keeps the power consumption at the same level in the worst case. Considering that the temperature will become an important problem for the high-throughput workloads, the PHA-WB can be effectively used to control the temperature.

Finally, our technique also improves the performance by reducing the average delay of read operations. We analyze how our technique influences the average read delay (duration of time between when the AMB assigns a read operation and when the data is available in the DRAM). Figure 7 shows the reduction in average read operation delay for each design. The average delay is reduced by up to 14.6% (9.1% on average across all benchmarks) for the 64-entry PHA-WB and up to 10.0% (6.6% on average across all benchmarks) for the 16-entry PHA-WB.

We further analyzed the impact of reducing DRAM access latency on system performance using M5. Starting from a baseline DRAM delay of 250 cycles, we adjust the average DRAM access delay based on the experiments depicted in Figure 7. Then, we simulate the performance of these workloads with these new DRAM access latencies. Figure 8 compares the average instruction per cycle (IPC) among 4 processor cores for these workloads.

Note that, compared to the 64-entry PHA-WB, the TAP reduces power consumption of W8 by 4.43%. Therefore, the TAP could achieve a higher power and temperature reduction compared with a fixed sized PHA-WB as shown by our experimental results. On the other hand, the average IPC is increased by up to 7.1% (3.4% on average across all benchmarks) for the 64-entry PHA-WB, by 2.9% on average for the 32-entry PHA-WB, and by 2.4% on average for the 16-entry PHA-WB. The average improvement in IPC for the TAP is 2.7 %. Hence, when compared to the biggest fixed sized PHA-WB, the TAP makes a trade-off between performance improvement and reduction in power consumption.

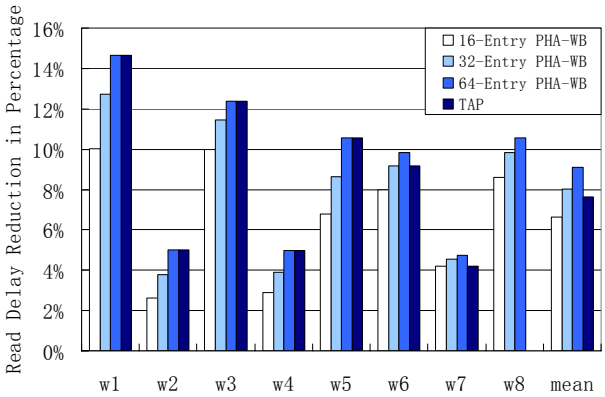

Figure 7. Reduction of average read delay for different workloads with different PHA-WB

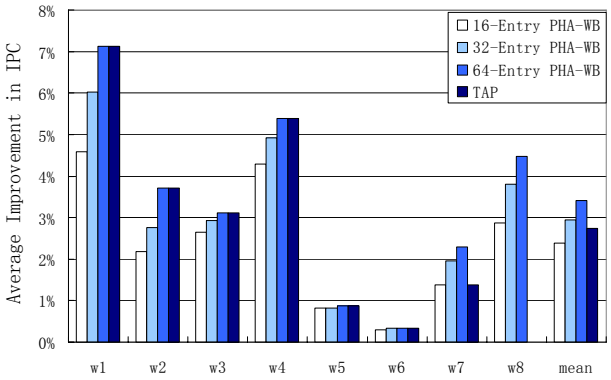

Figure 8. Improvement of average IPC for different workloads with different PHA-WB

6. CONCLUSIONS

We have proposed and evaluated an adaptive power and thermal management scheme for DRAM systems. Our proposed technique achieves power and temperature reduction by improving DRAM page hit rate with a write buffer. Since we do not delay read operations, our approach does not hurt the system performance. In fact, higher page hit rate improves the system performance. Our proposed technique could achieve power savings by as much as 18.36% (8.64% on average) and reduce temperature by up to 5.10°C (1.93°C on average).

Our proposed technique does not pose any limitations on the use of any other types of idle mode power optimization schemes or thermal-emergency intervention methods. Such techniques can co-exist along with our proposed DRAM architecture. Our architecture would provide the first level of optimization at no performance cost and further aggressive optimizations or interventions could be applied by trading-off performance.

7. ACKNOWLEDGEMENTS

We would like to thank the anonymous reviewers for their helpful comments. This work is in part supported by DOE Award DEFG02-05ER25691, NSF Awards CNS-0546305, CCF-0747201, CNS-0720691, IIS-0613568, CCF-0541337,

CNS-0551639, IIS-0536994, CCF-0444405. Gokhan Memik's work is also in part supported by Wissner-Slivka Chair funds.

8. REFERENCES

1. Binkert, N.L., R.G. Dreslinski, L.R. Hsu, K.T. Lim, A.G. Saidi, and S.K. Reinhardt, *The M5 simulator: modeling networked systems*. IEEE Micro, 2006. **26**(4): p. 52-60.
2. Delaluz, V., M. Kandemir, N. Vijaykrishnan, A. Sivasubramaniam, and M.J. Irwin, *Hardware and Software Techniques for Controlling DRAM Power Modes*. IEEE Transactions on Computers, 2001. **50**(11).
3. Fan, X., C.S. Ellis, and A.R. Lebeck, *Memory Control Policies for DRAM Power Management*, in *ISLPED'01*. 2001.
4. Iyer, J., 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.
5. JEDEC, *FBDIMM Specification: DDR2 SDRAM Fully Buffered DIMM (FBDIMM) Design Specification http://www.jedec.org/download/search/JESD2051.pdf*.
6. JEDEC, *FBDIMM: Advanced Memory Buffer (AMB) http://www.jedec.org/download/search/JESD82-20.pdf*.
7. Lebeck, A.R., X. Fan, H. Zeng, and C. Ellis, *Power Aware Page Allocation*, in *ASPLOS-IX*. 2000.
8. Lee, K.-B., T.-C. Lin, and C.-W. Jen, *An efficient quality-aware memory controller for multimedia platform SoC*. IEEE Transactions on Circuits and Systems for Video Technology,, 2005. **15**(5).
9. Lin, J., H. Zheng, Z. Zhu, H. David, and Z. Zhang, *Thermal Modeling and Management of DRAM Memory Systems*, in *ISCA'07*. 2007.
10. Lin, J., H. Zheng, Z. Zhu, Z. Zhang, and H. David, *DRAM-level prefetching for fully-buffered DIMM: design, performance and power saving*, in *ISPASS'07*. 2007.
11. Micron, *Calculating Memory System Power for DDR2*.
12. Micron, *DDR2 SDRAM FBDIMM http://download.micron.com/pdf/datasheets/modules/ddr2/HTF18C128_256x72FD.pdf*.
13. Micron, *DDR2 SDRAM http://download.micron.com/pdf/datasheets/dram/ddr2/512MbDDR2.pdf*.
14. Micron, *System Power Calculator, http://www.micron.com/support/designsupport/tools/powercalc/powercalc.aspx*.
15. Rambus, *RDRAM*, in *www.rambus.com*.
16. Rixner, S. *Memory Controller Optimizations for Web Servers*. in *MICRO-37*. 2004.
17. Shivakumar, P. and N.P. Jouppi, *CACTI 3.0: An Integrated Cache Timing, Power, and Area Model* WRL Research Report.
18. www.spec.org, *Standard Performance Evaluation Corporation. SPEC CPU2000* .