Thermal Modeling and Management of DRAM Memory Systems

Jiang Lin[1], Hongzhong Zheng[2], Zhichun Zhu[2], Howard David[3] and Zhao Zhang[1]

[1]Department of Electrical and Computer Engineering
Iowa State University
Ames, IA 50011
{linj,zzhang}@iastate.edu

[2]Department of Electrical and Computer Engineering
University of Illinois at Chicago
Chicago, IL 60607
{hzheng2,zzhu}@uic.edu

[3]Digital Enterprise Group
Intel Corp.
Hillsboro, OR 97124
howard.david@intel.com

ABSTRACT

With increasing speed and power density, high-performance memories, including FB-DIMM (Fully Buffered DIMM) and DDR2 DRAM, now begin to require *dynamic thermal management* (DTM) as processors and hard drives did. The DTM of memories, nevertheless, is different in that it should take the processor performance and power consumption into consideration. Existing schemes have ignored that. In this study, we investigate a new approach that controls the memory thermal issues from the source generating memory activities – the processor. It will smooth the program execution when compared with shutting down memory abruptly, and therefore improve the overall system performance and power efficiency. For multicore systems, we propose two schemes called *adaptive core gating* and *coordinated DVFS*. The first scheme activates clock gating on selected processor cores and the second one scales down the frequency and voltage levels of processor cores when the memory is to be overheated. They can successfully control the memory activities and handle thermal emergency. More importantly, they improve performance significantly under the given thermal envelope. Our simulation results show that adaptive core gating improves performance by up to 23.3% (16.3% on average) on a four-core system with FB-DIMM when compared with DRAM thermal shutdown; and coordinated DVFS with control-theoretic methods improves the performance by up to 18.5% (8.3% on average).

Categories and Subject Descriptors: B.3.2 [Primary Memory]: Design Styles

General Terms: Design, Management, Performance

Keywords: Thermal Modeling, Thermal Management, DRAM Memories

1. INTRODUCTION

With the wide employment of multicore processors, the capacity and speed of DRAM main memory systems must be improved to match the increasing demand from multiple processor cores. For high-performance DRAM memories, thermal control has become a realistic issue as it had for processors and hard drives [15]. Using cooling features such as fans over DRAM chips will increase the cooling budget and overall cost but may not fully address this issue. Additionally, systems like notebook computers and blade servers have limited space for installing new cooling components.

Recently, simple dynamic thermal management (DTM) techniques have been applied in notebook computers with DDR2 memories. Two simple DTM schemes have been used so far for DRAM memories: thermal shutdown and memory bandwidth throttling. Upon detected overheating of DRAM chips, with thermal shutdown, the memory controller stops all memory transactions and shuts down the DRAM chips until they are cooled down. With memory bandwidth throttling, the memory controller lowers bandwidth to reduce DRAM activities. However, abrupt thermal shutdown or bandwidth throttling will make the program execution fluctuating. Intuitively, the program execution is far away from optimal for a given thermal envelope: Thermal shutdown frequently stops the memory subsystem and consequently forces the processor to stall; and simple memory bandwidth throttling reduces the memory throughput while the processor runs at high speed. Furthermore, the power efficiency of the whole system including the processor, power supply and other components will not be optimal.

In this study, we take a new approach that controls the memory throughput by directly controlling the source that generates memory activities – the processor, when the memory thermal envelope is approached. We propose two new schemes and evaluate their effectiveness on systems with multicore processors and Fully Buffered DIMM (FB-DIMM) memories [8]. The first scheme, *Adaptive Core Gating*, applies clock gating on selected processor cores according to DRAM thermal state. The second scheme, *Coordinated DVFS* (dynamic voltage and frequency scaling), scales down the frequency and voltage levels of all processor cores, when the memory is to be overheated. Both schemes maintain the memory throughput as high as allowed by the current thermal limit; and therefore improve the average memory performance. Adaptive core gating further reduces L2 cache conflicts, which leads to lower memory traffic and fewer DRAM bank conflicts. It improves the performance of multiprogramming workloads of SPEC2000 programs by up to 23.3% (16.3% on average) on a four-core processor when compared with the simple thermal shutdown for a configuration used

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
ISCA'07, June 9–13, 2007, San Diego, California, USA.
Copyright 2007 ACM 978-1-59593-706-3/07/0006 ...$5.00.

in our study. Coordinated DVFS also reduces memory traffic slightly because the processor generates fewer speculative memory accesses when running at a lower frequency. In addition, the processor power efficiency is improved with voltage scaling. The scheme improves performance by up to 15.3% (3.4% on average), and may save the processor energy consumption by 37.4% on average, compared with the simple thermal shutdown.

We further use a PID (Proportional-Integral-Differential) method based on formal control theory to improve the efficiency of the proposed DTM schemes. It can make the system temperature to converge quickly to the target temperature, and further improve the performance of adaptive core gating by up to 25.6% (21.2% on average) and coordinated DVFS by up to 18.5% (8.3% on average) when compared with the simple thermal shutdown.

We have also presented a thermal and power model for FB-DIMM. FB-DIMM is designed for multicore processors to meet their demand of high bandwidth and large capacity. However, it has thermal issues when running at the peak performance for a while (usually less than a hundred seconds). It uses narrow and high-speed memory channels, and includes Advanced Memory Buffer (AMB) to buffer and transfer data between memory channels and DDR2 DRAM chips. In FB-DIMM, both the AMBs and DRAM chips may be overheated. The power density of an AMB can be as high as $18.5 Watt/cm^2$ [19]. To model the power consumption and thermal behavior of FB-DIMM, our model uses two sets of formulas: one by Micron Technology, Inc. for DRAM power consumption [21] and the other by Intel Corp. for AMB and DRAM temperature estimation in a stable state [14]. The heat generated by the AMBs and DRAM chips is determined by the memory throughput. The model estimates the dynamic change of temperatures of the AMBs and DRAM chips using the current memory throughput, which is collected by simulation.

In summary, we have made the following contributions in this paper:

- We propose two new DTM schemes for DRAM memories, which provide better performance than existing, simple DTM schemes. A new approach is used that controls the memory throughput from the source generating memory activities – the processor.

- We combine the proposed DTM schemes with a control-theoretic method, which further improves the performance by allowing the memory temperature to stay close to the target temperature without violating its thermal limit.

- We build a power and thermal model for FB-DIMM by using existing power and temperature estimation methods used by industry. We have also built a simulator to incorporate the model and to evaluate the proposed DTM schemes.

The rest of this paper is organized as follows. Section 2 introduces the power and thermal model for FB-DIMM. Section 3 describes the existing and proposed DTM schemes for DRAM main memory. Section 4 describes the experimental environment and Section 5 presents the results of our experiments. After discussing the related work in Section 6, we summarize this study in Section 7.

2. THERMAL MODELING OF FB-DIMM

In this study, we focus on systems with FB-DIMM as the main memory. FB-DIMM is a new type of main memory systems that can support both large capacity and high bandwidth. These two advantages make it suitable for high-performance computing.

2.1 Basic Structure of FB-DIMM

The structure of FB-DIMM [10] is shown in Figure 1. The introduction of Advanced Memory Buffer (AMB) makes the FB-DIMM system more likely to have thermal emergency. FB-DIMM has a two-level interconnect structure, the FB-DIMM channel and the DDR2 buses on the DIMMs. The AMB is a key component in this interconnect structure. The memory controller links to those AMBs through a narrow but high frequency point-to-point bus, forming a daisy chain. Figure 1 shows only one channel connected to the memory controller; in real systems, multiple channels can be connected to a single controller. The DRAM chips on a DIMM are connected to the DIMM's AMB; they are not directly connected to the channel bus.

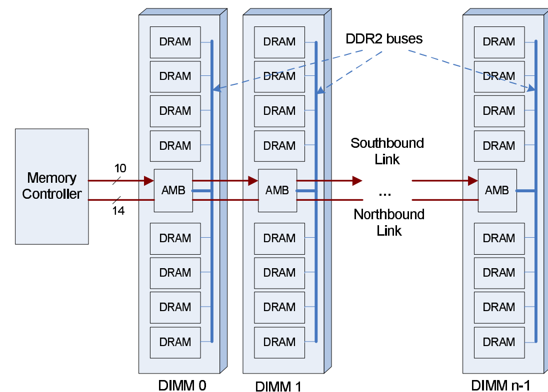

Figure 1: The structure of Fully-Buffered DIMM with one channel, n DIMMs and eight DRAM chips per DIMM. The memory controller is able to connect up to six channels, and each channel may connect up to eight DIMMs.

2.2 Power Model of FB-DIMM

We first develop a power model of FB-DIMM, including its DRAM chips and AMBs (with DDR2 bus interconnect). Based on the power model, we will develop a thermal model in Section 2.3. We assume that the FB-DIMM uses the close page mode with auto precharge. This configuration achieves better overall performance in multicore program execution than open page mode or close page mode without auto precharge. We also assume that the FB-DIMM uses 1GB DDR2-667x8 DRAM chips made by 110nm process technology. Additionally, the memory access burst length is fixed at four to transfer a single L2 cache block of 64 bytes over two FB-DIMM channels.

A Simple DRAM Power Model. We derive a simple power model from a DRAM power calculator [21] provided by Micron Technology, Inc. The DRAM power at a given moment is estimated as follows:

$$P_{\text{DRAM}} = P_{\text{DRAM_static}} + \alpha_1 \times \text{Throughput}_{\text{read}} + \alpha_2 \times \text{Throughput}_{\text{write}} \quad (1)$$

We assume that the DRAM does not enter low power modes and on average during 20% of time the DRAM banks of a DIMM are all precharged. This is a representative setting and is used as the default one by the power calculator. With those assumptions, the DRAM static power can be estimated as a constant for a relatively long time interval, e.g. a few milliseconds[1]. The value is 0.98 Watt for a single FB-DIMM, derived by the DRAM power calculator. In the calculator, this value includes the power for DRAM refreshing, although that part is actually dynamic power consumption.

The second and third components belong to the dynamic DRAM power consumption, and are determined by the read throughput, write throughput and row buffer hit rate. With the close page mode and auto-precharge, each DRAM read or write causes three DRAM operations: row activation (RAS), column access (CAS) and precharge (PRE). Each row activation consumes the same amount of energy, and so does each precharge. A column access of a read, however, consumes slightly less power than that of a write. The row buffer hit rate is zero with close page mode and auto-precharge, therefore it does not appear in Equation 1. The value of α_1 is 1.12 Watt/(GB/s) and that of α_2 is 1.16 Watt/(GB/s) for a single FB-DIMM, derived from the DRAM power calculator. Finally, the read and write throughput are collected in the simulation.

AMB Power Modeling. To calculate the AMB power consumption, we first discuss how AMB works. The FB-DIMM channel interconnect has two unidirectional links located in the AMBs, southbound link and northbound link, which operate independently. The southbound link carries commands and data to be written; and the northbound link carries the read data returned from the DIMMs. As shown in Figure 2, the AMB is a small logic component attached to each DIMM and sits between the memory controller and DRAM chips. It receives commands and data from the FB-DIMM bus; and then determines whether the commands and data are for its memory devices or not. If the answer is yes, the AMB translates the commands and data to the internal DDR2/DDR3 format; otherwise, it will forward the commands and data to the next AMB or the memory controller through the FB-DIMM channel.

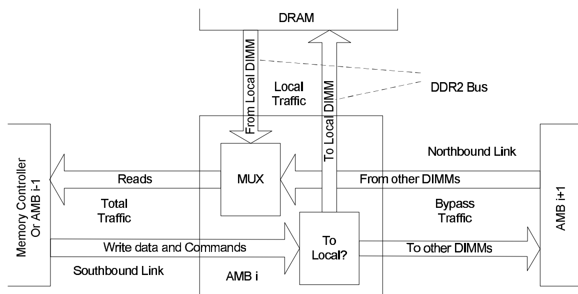

Figure 2: Four categories of data traffic that flows through AMB.

An AMB consumes energy in each local request (directed to the local DRAMs), and in each bypassed request (to other DIMMs). For each local read request, the AMB consumes energy in decoding and forwarding the commands to the local DDR2 bus, and then receiving the read data and sending them back through the FB-DIMM channel. For each local write, the AMB decodes the commands from FB-DIMM channel, and then sends them with data through the local DDR2 bus. For each bypassed read request, the AMB passes the commands through the southbound link and later passes the data through the northbound link. For each bypassed write request, the AMB passes the command and data through the southbound link. The number of commands and the amount of data transferred are the same for a read or a write request. Therefore, we assume that each local read or write request consumes the same amount of energy, and so does each bypassed read or write request. A local request consumes more energy than a bypassed request.

Based on the above analysis, we model the AMB power consumption as a linear function of memory throughput of bypass traffic and local traffic:

$$P_{\text{AMB}} = P_{\text{AMB_idle}} + \beta \times \text{Throughput}_{\text{Bypass}} + \gamma \times \text{Throughput}_{\text{Local}} \quad (2)$$

$P_{\text{AMB_idle}}$ represents the power consumption when there is no memory traffic presented to AMB. We derive the values of $P_{\text{AMB_idle}}$ and coefficients β and γ from Intel specification [14] for FB-DIMM. The values are shown in Tables 1. $P_{\text{AMB_idle}}$ has two possible values, 4.0 Watts for the last AMB of an FB-DIMM channel and 5.1 Watts for other AMBs. The difference exists because the memory controller and the AMBs must keep in synchronization all the time, which consumes power, while the last AMB only needs to synchronize with one side. The bypass and local throughout is collected in the simulation.

Parameters	Value
$P_{\text{AMB_idle}}$ (last DIMM)	4.0 watt
$P_{\text{AMB_idle}}$ (other DIMMs)	5.1 watt
β	0.19 watt/(GB/s)
γ	0.75 watt/(GB/s)

Table 1: The values of parameters in Equation 2 for FB-DIMM with 1GB DDR2-667x8 DRAM chips made by 110nm process technology.

2.3 Thermal Model of FB-DIMM

We build a simple thermal model for FB-DIMM based on the power model above. First of all, because the DIMMs in FB-DIMM memory are “far” from each other and cooling air flow passes through the space between them, we assume that there is no thermal interaction between any two DIMMs. The focus is the thermal behavior of a single DIMM, including the thermal interactions between the DRAM chips and the AMB. Our analysis is based on a previous analysis done by Intel [20], which models the stable temperature of FB-DIMM. Our model extends to the dynamic temperature of FB-DIMM.

We first describe the modeling of stable temperatures of the AMB and DRAMs, i.e. the temperatures if the memory throughput does not change. For a general physical system with heat source and sink, the stable temperature is the balance point where the heat generating speed equals to the heat dissipation speed. The higher the temperature, the

[1]If all DRAM banks of a DIMM are precharged, the static power is lower than otherwise by a small margin.

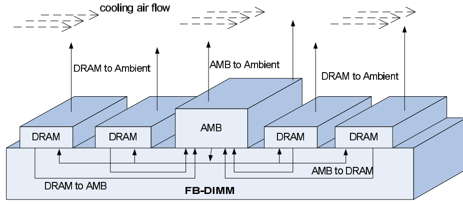

Figure 3: Heat dissipation of FB-DIMM. The arrows represent heat dissipation paths.

Heat spreader type	AOHS (on AMB)			FDHS (on DIMM)		
Air velocity(m/s)	1.0	**1.5**	3.0	**1.0**	1.5	3.0
$\Psi_{\rm AMB}$(°C/W)	11.2	**9.3**	6.6	**8.0**	7.0	5.5
$\Psi_{\rm DRAM_AMB}$(°C/W)	4.3	**3.4**	2.2	**4.4**	3.7	2.9
$\Psi_{\rm DRAM}$(°C/W)	4.9	**4.0**	2.7	**4.0**	3.3	2.3
$\Psi_{\rm AMB_DRAM}$(°C/W)	5.3	**4.1**	2.6	**5.7**	4.5	2.9
$\tau_{\rm AMB}$(seconds)	50					
$\tau_{\rm DRAM}$(seconds)	100					

Table 2: The value of parameters in the thermal model for the AMB and DRAM chips in the given type of FB-DIMM used in our simulation. The columns in bold type are used in our experiments.

faster the heat dissipation speed. Figure 3 shows the heat dissipation paths in a single DIMM. The heat generated by AMB is dissipated by two paths: one to the heat spreader and then to ambient, and another down to the raw card (DIMM board). Similarly, the heat from each DRAM chip has these two dissipation paths, but may or may not have the heat spreader in the first path. Thermal interactions exist among the AMB and DRAMs through the raw card.

The AMB and DRAMs have different stable temperatures[2] that are affected by several factors. First, the heat generation of the AMB and DRAM is determined by the memory throughput. Second, the higher the velocity of the cooling air flow, the quicker the heat dissipation from the AMB and DRAMs to the ambient. Third, the type of heat spreader can change the distribution of heat dissipation between the two paths. There are two types of heat spreader for FB-DIMM: AMB Only Heat Spreader (AOHS) and Full DIMM Heat Spreader (FDHS) [19]. The AOHS only contacts and covers the AMB. The FDHS covers the full length of the DIMM including the AMB and DRAMs, providing another channel for the thermal interactions between AMB and DRAMs. One can expect that the difference between stable AMB temperature and the DRAM temperature of DIMMs with FDHS is smaller than that with AOHS. Finally, the ambient temperature affects the stable temperatures: The higher the ambient temperature, the higher the stable temperatures.

We use the following two equations to calculate the stable temperatures, which are simplified versions of the Intel's study [20].

$$T_{\rm AMB} = T_{\rm A} + P_{\rm AMB} \times \Psi_{\rm AMB} + P_{\rm DRAM} \times \Psi_{\rm DRAM_AMB} \quad (3)$$

$$T_{\rm DRAM} = T_{\rm A} + P_{\rm AMB} \times \Psi_{\rm AMB_DRAM} + P_{\rm DRAM} \times \Psi_{\rm DRAM} \quad (4)$$

Parameter $T_{\rm A}$ is the ambient temperature. Parameter $\Psi_{\rm AMB}$ is the thermal resistance from the AMB to the ambient; thermal resistance is the ratio of the change of stable temperature over the change of power consumption. $\Psi_{\rm DRAM}$ is the thermal resistance from a DRAM chip to the ambient. Parameters $\Psi_{\rm AMB_DRAM}$ and $\Psi_{\rm DRAM_AMB}$ are the thermal resistances from AMB to DRAM and from DRAM to AMB, respectively. The power density and heat generation of the AMB are much higher than those of the DRAM. Therefore, we are only concerned with the chip(s) next to the AMB, which has the highest temperature. The values of those parameters are from the Intel's study and listed in Table 2. To limit the experimental time, we choose two cooling configurations in this study: *AOHS+1.5m/s* (AOHS_1.5) and *FDHS+1.0m/s* (FDHS_1.0).

[2]The AMB has a higher thermal limit than the DRAMs.

We now model the dynamic temperature changes with varying memory throughput in program execution. We use the following equation to describe the dynamic temperature:

$$T(t + \triangle t) = T(t) + (T_{\rm stable} - T(t))(1 - {\rm e}^{-\frac{\triangle t}{\tau}}) \quad (5)$$

Basically, the equation treats the temperature in a thermal system like the voltage in an electrical RC circuit. This idea has been used in previous studies [23, 24] and the formula above is based on a classic equation for the electrical RC circuit [11]. In this equation, τ is the time for the temperature difference to be reduced by $1/e$, i.e. $T(t+\tau) - T(t) = (1 - 1/e)(T_{\rm stable} - T(t))$, if the heat generation rate is a constant. We obtain the value of τ for the AMB and DRAMs by observing their temperature changes in a physical testing environment using the same type of FB-DIMM as in our simulation environment. It is rounded to an integer number of seconds.

Because the leakage power is negligible for DRAM devices and AMBs, we do not include the thermal-leakage feedback loop in the equation. In other words, we assume their leakage power rate does not increase with the temperature. In an experimental testbed of FB-DIMM memory subsystem, we observed only 2% increase of power rate as the DRAM subsystem heated up. Additionally, the model can be adapted to other DRAMs because the power profiles of various DRAMs are fairly consistent, both across manufacturers and across generations.

3. DYNAMIC THERMAL MANAGEMENT FOR FB-DIMM MEMORY

In this section, we first discuss existing DTM schemes for main memory, and then describe our DTM schemes and the use of formal control method. All DTM schemes assume that thermal sensors are used to monitor the DRAM temperature; and for FB-DIMM, the AMBs have already integrated thermal sensors.

3.1 Existing Memory DTM Schemes

In *thermal shutdown*, the memory controller (or the operating system) periodically reads the temperature of DRAMs from the thermal sensors. The period may be a fraction of second. If the temperature exceeds a preset thermal threshold, the memory controller stops all accesses to the DRAMs. The controller keeps checking the temperature periodically and resumes DRAM accesses when the temperature drops

below the threshold by a preset margin. In *bandwidth throttling* [20], multiple thermal emergency levels are used to indicate how close the DRAM temperature is to the preset threshold. The BIOS (or the memory controller or OS) periodically reads the temperature, evaluates the thermal emergency level, and decides a memory traffic limit for the current period. Then, the memory controller will enforce this traffic limit. In the rest of this paper, we refer these two schemes as *DTM-TS* and *DTM-BW*, respectively.

3.2 Proposed DTM Schemes

We propose *adaptive core gating (DTM-ACG)* and *coordinated dynamic voltage and frequency scaling (DTM-CDVFS)* schemes. The two schemes are designed for multicore processors. Unlike DTM-TS and DTM-BW that control memory throughput locally at the memory side, the two schemes directly control the multicore processor to affect the memory throughput. For a processor of N cores, DTM-ACG may shut down 1 to N cores adaptively according to the current thermal emergency level. The core shutdown is to apply clock gating, i.e. stop the clock signal to the specific core. To ensure fairness among benchmarks running on different cores, the cores can be shut down in a round-robin manner. By shutting down some cores, memory throughput is expected to decrease and so is the DRAM and AMB heat generation rate. DTM-CDVFS may lower the frequency and voltage levels of all cores according to the DRAM/AMB thermal emergency level. In other words, it directly links the DRAM/AMB thermal level to the processor frequency and voltage level. In the highest thermal emergency level, for both DTM-ACG and DTM-CDVFS, the memory will be fully shut down. The two schemes may be implemented in OS or memory controller.

Both schemes may make the program execution running more smoothly than DTM-TS and DTM-BW, which shut down the memory system or reduce the bandwidth without considering the processor execution. DTM-ACG has another advantage for multicore processors with shared L2/L3 caches: By reducing the number of active cores, it reduces L2/L3 cache contention and therefore the total number of cache misses. Consequently, the total amount of memory traffic will be reduced and less heat will be generated. DTM-CDVFS has another advantage of its own: It may improve the processor energy efficiency significantly by *proactively* putting the processor in a power mode in coordination with the current DRAM thermal limit. With DTM-BW, a passive DVFS policy at the processor side will not respond timely because of the relatively long delay in power mode switch with DVFS. With DTM-CDVFS, however, the processor power mode will be switched proactively when the change of memory throughput limit is foreseen.

3.3 DTM-ACG and DTM-CDVFS Integrated with Formal Control Method

We further apply a formal control theory method called PID (Proportional-Integral-Differential) into DTM-ACG and DTM-CDVFS schemes. The PID method has recently been used in the processor thermal control [23, 24, 26, 27, 6]. A PID controller uses the following equation:

$$m(t) = K_c \left(e(t) + K_I \int_0^t e(t)dt + K_D \frac{de}{dt} \right) \quad (6)$$

The equation has three components on the right-hand side: the proportional factor, the integral factor and the differential factor. At any time t, $e(t)$ is the difference between the target temperature and the measured temperature; K_c, K_I and K_D are proportional, integral and differential constants that are tuned for the specific system; and proper control actions will be taken according to the controller output $m(t)$. The control action is application-dependent; for example, to set the processor frequency according to the range of $m(t)$. The setting of the ranges and the mapping of each range to a control decision are also application-dependent. For DTM-ACG, the control action is to set the number of active processor cores. For DTM-CDVFS, the control action is to set the processor frequency and voltage levels. We use two PID controllers, one for the AMB thermal control and another for the DRAM thermal control. For any given configuration that we have studied, either DRAM or AMB is always the thermal limit during program execution. The action by the corresponding PID controller will be taken.

The advantages of using the PID formal controller in thermal control is two-fold: First, the robust PID controller may make the temperature to converge to the target temperature within a guaranteed time limit; and the target temperature can be set close to the thermal limit to minimize the performance loss. Second, by taking into account of the history information in the integral factor and the future prediction in the differential factor, the PID controller can smooth the application running by proper control decisions from quantifying the temperature feedback [23, 6].

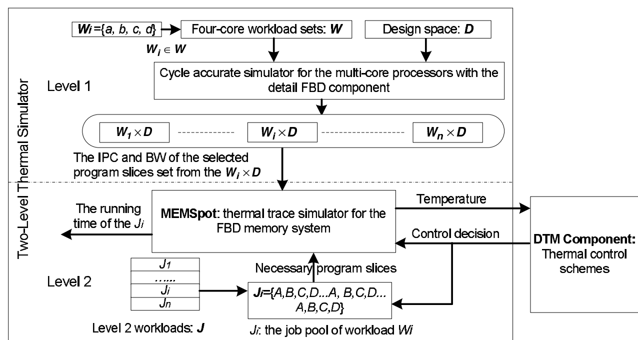

Figure 4: Two-level thermal simulator.

4. EXPERIMENTAL METHODOLOGY

4.1 Two-Level Thermal Simulator

It takes relatively long time for the AMB and DRAM to overheat, usually tens of seconds to more than one hundred seconds[3]. Therefore, we need to evaluate the DRAM DTM schemes for at least thousands of seconds. Direct cycle-accurate simulation for studying DRAM thermal management is almost infeasible at this time length. To address this issue, we propose and implement a two-level simulation infrastructure as shown in Figure 4. The first-level is a cycle-accurate architectural simulator, which is used to build traces with performance and memory throughput information for all possible running combinations of workloads under each DTM design choice. The second-level simulator

[3]By comparison, a processor may overheat in tens of milliseconds.

Parameters	Values
Processor	4-core, 4-issue per core, 21-stage pipeline
Clock frequency scaling	3.2GHz at 1.55V, 2.8GHz at 1.35V, 1.6GHz at 1.15V, 0.8GHz at 0.95V
Functional units	4 IntALU, 2 IntMult, 2 FPALU, 1 FPMult
ROB and LSQ size	ROB 196, LQ 32, SQ 32
Branch predictor	Hybrid, 8k global + 2K local, 16-entry RAS, 4K-entry and 4-way BTB
L1 caches (per core)	64KB Inst/64KB Data, 2-way, 64B line, hit latency: 1 cycle Inst/3-cycle Data
L2 cache (shared)	4MB, 8-way, 64B line, 15-cycle hit latency
MSHR entries	Inst:8, Data:32, L2:64
Memory	2 logic (4 physical) channels, 4 DIMMs/physical channel, 8 banks/DIMM
Channel bandwidth	667MT/s (Mega Transfers/second), FB-DIMM-DDR2
Memory controller	64-entry buffer, 12ns overhead
Cooling configuration	AOHS with 1.5m/s cooling air velocity and FDHS with 1.0m/s cooling air velocity
DTM parameters	DTM interval 10ms, DTM control overhead 25μs, DTM control scale 25%
Major DRAM parameters	(5-5-5) : active to read tRCD 15ns, read to data valid tCL 15ns, precharge to active tRP 15ns
Other DRAM parameters	tRAS=39ns, tRC=54ns, tWTR=9ns, tWL=12ns, tWPD=36ns, tRPD=9ns, tRRD=9ns

Table 3: Simulator parameters.

emulates the power and thermal behavior of memory systems using those traces. The traces use 10ms time window, which is sufficient to capture the fluctuation of temperature. DRAM temperature fluctuates slowly, up to two degrees Celsius per second as we observed on real machines.

As for the first-level simulation, we use M5 [1] as the base architectural simulator and extend its memory part to include a memory simulator for multi-channel FB-DIMM with DDR2 DRAM devices. The details of FB-DIMM northbound and southbound links and isolated command and data buses inside FB-DIMM are simulated, and so are DRAM access scheduling and operations at all DRAM chips and banks. Table 3 shows the major parameters of the pipeline, the memory system, the DTM techniques and the DRAM operations. The outputs of the simulator are the traces of the processor performance and memory throughput of each workload W_i under the entire explored design space D, including varied memory bandwidth, processor running speed and voltage level, and number of active processor cores. The set of all traces $W_i \times D$ is then fed into the second-level simulator for power and thermal simulation.

The second-level simulator, MEMSpot, uses the power and thermal models described in Section 2 to emulate the power and thermal behavior of the DRAM chips and AMBs in the FB-DIMM memory system. The memory throughput values used in the models are provided by the first-level simulator. The values of other parameters are given in Section 2. The MEMSpot simulates the change of DRAM/AMB temperatures using those parameters for the current processor running mode, e.g. the frequency and voltage level. The temperature data are used by the DTM component, which makes control decisions and informs the MEMSpot any changes of processor running mode.

4.2 Workloads

Each processor core is single-threaded and runs a distinct application. From the SPEC2000 benchmark suite [25], we select twelve applications that require high memory bandwidth when the four-core system runs four copies of the application. Eight of them get memory throughput higher than 10GB/s, *swim, mgrid, applu, galgel, art, equake, lucas* and *fma3d*. The other four get memory throughput between 5GB/s and 10GB/s, *wupwise, vpr, mcf* and *apsi*. Then we construct eight multiprogramming workloads randomly from these selected applications as shown in Table 4.

In order to observe the memory temperature characteristics in long run, the second-level simulator runs the multiprogramming workloads as batch jobs. For each workload W, its corresponding batch job J mixes multiple copies (fifty in our experiments) of every application A_i contained in the workload. When one application finishes its execution and releases its occupied processor core, a waiting application is assigned to the core in a round-robin way. In order to limit the simulation time of the first-level architectural simulator while still getting the accurate behavior of a program's execution, each application is approximated by replicas of a representative program slice of 100 million instructions picked up according to SimPoint 3.0 [22]. To determine the number of replicas for each application, we use the simulator sim-safe from the SimpleScalar 3.0 suite [4] to get the total number of instructions of each application and then divide it by 100 million. Using this approach, we are able to simulate the execution of a batch job with actual running time of thousands of seconds within a few days. This allows us to balance between the simulation accuracy and time, and to explore a wide design space of DTM schemes.

4.3 DTM Parameters

The thermal limits for the AMB and DRAM chips are 110°C and 85°C, respectively, for the FB-DIMM with 1GB DDR2-667x8 DRAM we choose in this study [14]. We define five thermal emergency levels, L1 to L5 for the DTM schemes as shown in Table 5. DTM-TS keeps the memory system turned on in states L1/L2 and keeps it shut down in state L5. As for states L3/L4, DTM-TS shuts down the memory system when the AMB temperature ever reaches 110.0°C and keeps it off until the temperature drops to 109.0°C; and similarly for the DRAM temperature. The control decisions by the DTM-BW, DTM-ACG and DTM-CDVFS schemes are self explained in the table. The DTM scale indicates the difference between any two control decisions next to each other.

4.4 Parameters in PID Formal Controller

In the PID formal controller, parameters K_c, K_I and K_D are generally obtained by heuristics and/or performance tuning. We use performance tuning and choose the following values: $K_c = 10.4$, $K_I = 180.24$, and $K_D = 0.001$ for AMB, and $K_c = 12.4$, $K_I = 155.12$ and $K_D = 0.001$ for DRAM. This approach is used in a previous study [23]. The PID con-

Workload	Benchmarks	Workload	Benchmarks
W0	swim, mgrid, applu, galgel	W1	art, equake, lucas, fma3d
W2	swim, applu, art, lucas	W3	mgrid, galgel, equake, fma3d
W4	swim, art, wupwise, vpr	W5	mgrid, equake, mcf, apsi
W6	applu, lucas, wupwise, mcf	W7	galgel, fma3d, vpr, apsi

Table 4: Workload mixes.

Thermal Emergency Level	L1	L2	L3	L4	L5	DTM scale
AMB Temp. Range (°C)	(-, 108.0)	[108.0, 109.0)	[109.0, 109.5)	[109.5, 110.0)	[110.0,-)	
DRAM Temp. Range (°C)	(-, 83.0)	[83.0, 84.0)	[84.0, 84.5)	[84.5, 85.0)	[85.0, -)	
DTM-TS: On/Off	On		On/Off		Off	100%
DTM-BW: Bandwidth	No limit	19.2GB/s	12.8GB/s	6.4GB/s	Off	25%
DTM-ACG: # of Active Cores	4	3	2	1	0	25%
DTM-CDVFS: Freq./Vol.	3.2GHz@1.55V	2.4GHz@1.35V	1.6GHz@1.15V	0.8GHz@0.95V	Stopped	25%

Table 5: Thermal emergency levels and their default settings used for the chosen FB-DIMM.

troller's target temperatures of the AMB and DRAMs are 109.8 and 84.8°C, respectively. In our FB-DIMM configuration, the setting leads to quick settling time and guarantees that the thermal limits will not be exceeded. To avoid the saturation effect [23, 6] created by the integral factor, we only turn on the integral factor when the temperature exceeds a certain threshold, 109.0°C for the AMB and 84.0°C for the DRAM by default; the integral factor is frozen when the control output saturates the actuator, which can effectively make the PID controller to respond quickly to temperature changes.

5. EFFECTIVENESS OF MEMORY DTM SCHEMES

5.1 Performance Impact of Thermal Release Point

With DTM-TS, when the temperature exceeds the TDP (thermal design point), thermal management mechanisms are triggered; after the temperature drops below the TRP (thermal release point), the mechanisms are disabled. For a given system, the choice of TRPs affects the degree of performance loss due to thermal management. According to the FB-DIMM specification [14], the TDPs of AMB and DRAM chips are 110.0°C and 85.0°C, respectively, for the FB-DIMM that we choose. In this section, we will first study the performance impact of TRPs in DTM-TS. The other schemes use more levels of thermal thresholds, and adjusting the thresholds shows similar impact.

Figure 5 shows the running time of workloads using DTM-TS with different TRP values under FDHS_1.0 (Full DIMM Heat Spreader with air velocity $1.0m/s$) and AOHS_1.5 (AMB Only Heat Spreader with air velocity $1.5m/s$) configurations. For comparison, the performance of an ideal system without any thermal limit (No-limit) is also presented. In the FDHS_1.0 configuration, the DRAMs usually enter thermal emergency before the AMBs, therefore we only vary the DRAM TRP. In the AOHS_1.5 configuration, the AMBs usually enter thermal emergency first, therefore we only vary the AMB TRP.

As shown in the figure, the performance loss due to thermal emergency is large. The running time of DTM-TS is up to three times of that without thermal limit. As expected, a higher TRP value causes smaller performance loss. For instance, compared with no thermal limit, the execution time of workload W1 is increased by 152% when the DRAM TRP is 81.0°C under FDHS_1.0, and the increase drops to 84% when the DRAM TRP is 84.5°C. A higher TRP value allows the system to stay at normal execution mode longer. In addition, the falling speed of temperature decreases as the temperature drops since the difference between the device and ambient temperatures is narrowing. As a result, high TRP values are desirable for performance purpose. However, we cannot set the TRP value of a component too close to its TDP value due to imperfect thermal sensors and delay on sensor reading. Thus, in the rest of experiments, we set the TRP values to 109.0°C for AMB and 84.0 for DRAM chips, respectively (1.0°C from their corresponding TDP values).

5.2 Performance Comparison of DTM Techniques

Running Time. Figure 6 presents the running time of the DTM schemes normalized to that of the ideal system without thermal limit. We do not present the data of DTM-TS with PID (the formal control method) because DTM-TS has only two control decisions and we find it does not benefit from the PID approach. The figure shows that the choice of DTM schemes affects the performance significantly: The normalized running time ranges from 0.97 to 2.41. Notice that all DTM schemes avoid thermal risk; and shorter running time means better performance.

The proposed DTM-ACG scheme has much better performance than DTM-TS and DTM-BW techniques; and the proposed DTM-CDVFS scheme is moderately better than those two. The use of PID further improves the performance of DTM-ACG, DTM-CDVFS and DTM-BW. With the AOHS_1.5 configuration (right in the figure), the average normalized running time of DTM-TS and DTM-BW is 1.82 and 1.81. DTM-ACG and DTM-CDVFS improve it to 1.52 and 1.75, respectively. The use of PID further improves it to 1.75, 1.46 and 1.68 for DTM-BW, DTM-ACG and DTM-CDVFS schemes, respectively. The performance with the FDHS_1.0 cooling package has a similar trend.

Under AHOS_1.5, the DTM-BW scheme has almost the same performance as DTM-TS. Compared with DTM-TS, DTM-ACG without PID can improve performance by up to 23.3% (for workload W1) and 16.3% on average; and

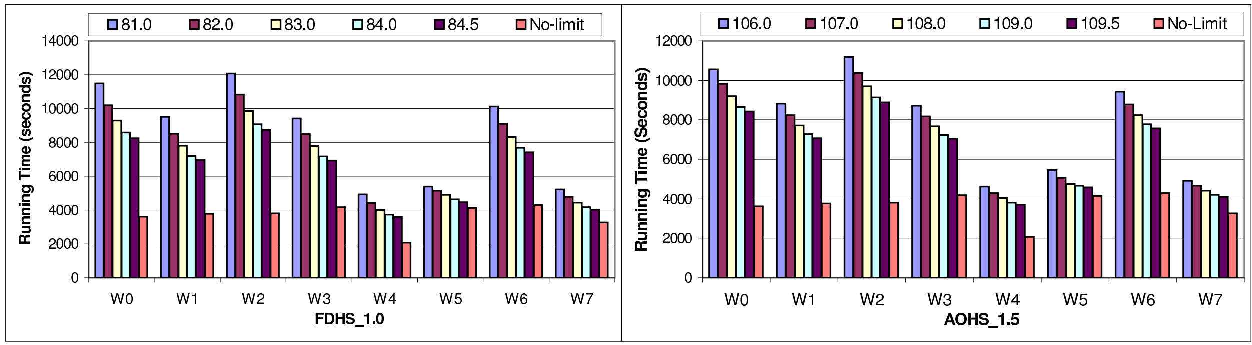

Figure 5: Performance of DTM-TS with varied DRAM TRP in the FDHS_1.0 cooling configuration (left) and AMB TRP in the AOHS_1.5 cooling configuration (right). The DRAM TDP is 85.0°C and the AMB TDP is 110.0°C.

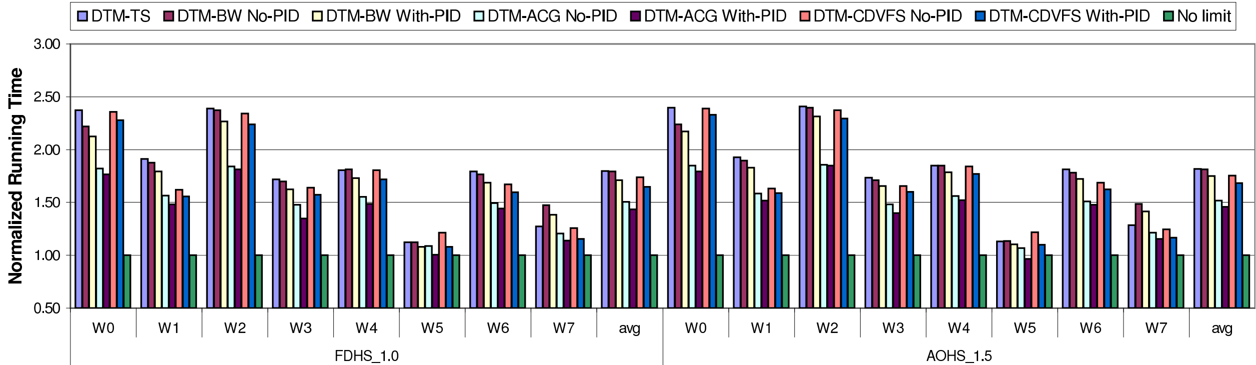

Figure 6: Normalized running time for DTM schemes.

DTM-CDVFS without PID can improve performance by up to 15.3% (for W2) and 3.4% on average. Combined with the PID method, the maximum performance improvement of DTM-ACG and DTM-CDVFS is 25.6% and 18.5%, respectively; and their average performance improvement is 21.2% and 8.3%, respectively. We will analyze the sources of performance gains in following discussion. It is worth noting that the performance of W5 when using DTM-ACG combined with PID is even better than that without thermal limit. A major reason is that the L2 cache conflicts drops when ACG is applied (miss rate dropping from 69.0% to 64.7% under AHOS_1.5).

Sources of Improvement. Next, we will analyze the sources of performance gains. We first look into the impact of DTM techniques on the total amount of memory traffic. Figure 7 shows the total memory traffic of those DTM schemes normalized to that of systems without memory thermal limit. As expected, the DTM-TS scheme does not affect the total memory traffic. The DTM-BW scheme throttles the memory bandwidth. It decreases the total memory traffic for workload W0; but increases the traffic for workload W7. For other workloads, its impact on memory traffic is not significant. We find that the L2 cache miss rate of W0 drops from 45.5% in DTM-TS to 40.6% in DTM-BW; and that of W7 increases from 25.3% in DTM-TS to 28.8% in DTM-BW. For other workloads, the differences of L2 cache miss rates are very small between DTM-TS and DTM-BW. We further find that the reason for the changes of L2 cache miss rates for those two particular workloads is the change of running time for different benchmark combinations. We leave this job scheduling issue to future work. The other workloads do not show this effect.

When the processor runs at a slower speed, it will generate fewer speculative memory accesses. Thus, the DTM-CDVFS scheme has the potential to reduce the memory traffic. On average, it reduces the traffic by 4.5% for both FDHS_1.5 and AOHS_1.0 configurations. The DTM-ACG scheme is the most effective in reducing the memory traffic, since it can reduce the amount of L2 cache conflicts when some of the processor cores are clock gated. It reduces the traffic for every workload; and the average traffic reduction is 16.7% for FDHS_1.5 and 17.0% for AOHS_1.0. When the control-theoretic method, PID, is applied, the total memory traffic is slightly increased. The reason is that it attempts to let the processor run at higher frequencies and with more active cores as long as the thermal limit is satisfied. Thus, the reduction on memory traffic is smaller.

The traffic reduction cannot fully explain the performance gain of PID control. The use of PID improves the overall performance with a slight increase of the memory traffic. In order to show other sources of performance improvement, in Figure 8, we present temperature curves of those DTM schemes for workload W0 under configuration AOHS_1.5 as predicted by the thermal model. Because the AMB instead of DRAM chips is expected to have thermal emergency under this configuration, only the AMB temperature is presented. The workload W0 contains four benchmarks demanding high memory bandwidth. The data show the AMB temperature changes during the first 1000 seconds of execution in one-second interval.

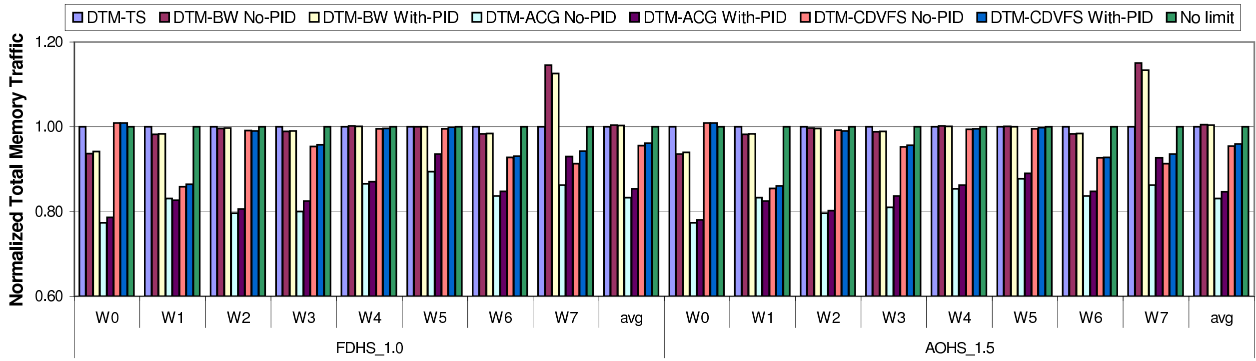

Figure 7: Normalized total memory traffic for DTM schemes.

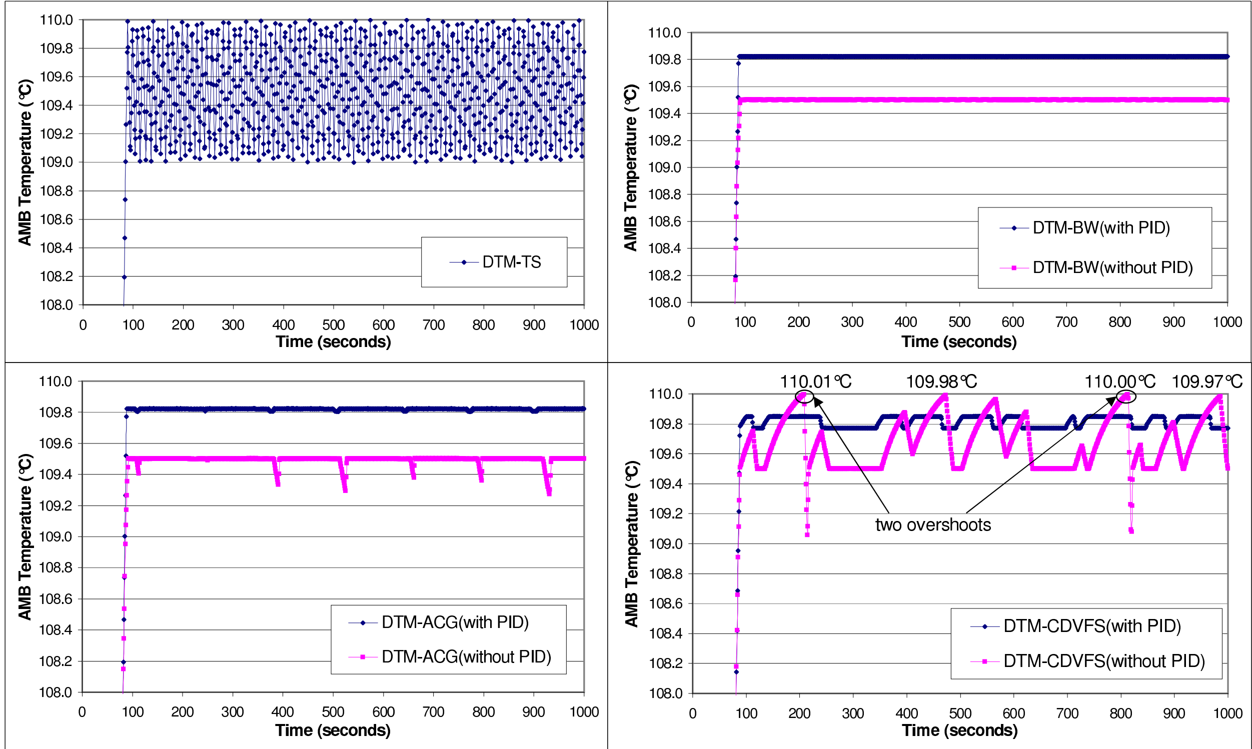

Figure 8: AMB temperature changes of DTM schemes for W0 with AOHS_1.5.

As expected, the AMB temperature swings between 109.0 and 110.0°C with DTM-TS, which is exactly defined by the scheme and thermal triggers. For DTM-BW without PID, the temperature swings around 109.5°C. This means that the memory bandwidth is throttled between 6.4GB/s and 12.8GB/s. We can see that one advantage of DTM-BW is that the AMB temperature is very stable and predictable. Thus, using this scheme, the temperature thresholds can be set very close to the thermal limit. When combined with the PID controller, the DTM-BW scheme makes the temperature to stick around 109.8°C. A higher stable temperature without violating thermal limit means that the system can stay at the normal execution mode longer, and thus can achieve better performance. For the three schemes, DTM-BW, DTM-ACG and DTM-CDVFS, combining with the PID method allows the AMB temperature to stay at a higher level than without PID. This is one of the reasons that the PID method can further improve performance for those DTM schemes.

For DTM-ACG without PID, most of time, the AMB temperature stays around 109.5°C and only one or two cores are active. The spikes of the curve indicate that during those periods, even with two active cores, the stable temperature is lower than 109.5°C. Thus, more cores could have been enabled. As shown in the figure, the use of PID eliminates almost all spikes. Additionally, we find from the simulation data (not shown here) that three or four cores are active during those periods. This is one of the reasons that the PID controller can improve performance.

For DTM-CDVFS without PID, most of time, the temperature swings between 109.5 and 110.0°C. Thus, its average temperature is higher than others. This is another source of performance gain for DTM-CDVFS. From the figure, we can see that the temperature reaches 110.0°C twice during the

1000 seconds period. Under such emergent cases, the memory is shut down until the AMB temperature drops below 109.0°C. The reach of the highest thermal emergency level (overshoot in the figure) is a potential thermal risk, which are eliminated by employing the PID controller. When DTM-CDVFS is combined with PID, the temperature sticks around 109.8°C and never overshoot. This allows us to set the target temperature of PID controller as high as 109.8°C. Without the PID controller, we must set the threshold lower to avoid overshoot. As mentioned earlier, the ability to stay at higher average temperature is another source of performance gains for the PID method.

5.3 Other Discussions

DTM Interval. In previous discussions, the memory temperature is checked for every 10ms. In general, a shorter DTM interval allows the thermal emergency to be handled more timely, especially when there is a danger of overshoot; while a longer interval has lower DTM overhead. We have done experiments on four DTM intervals: 1ms, 10ms, 20ms and 100ms. Due to space limit, we only summarize the results here. For all DTM schemes, the running time variance of these four different DTM intervals is within 4.0%. Since we assume that each DTM period has 25μs overhead, which accounts for 2.5% overhead for the DTM interval of 1ms, using this short interval causes longer running time than others. The variance of running time of the other three DTM intervals is within 2.0%. Based on these results, we believe 10ms is a good design choice for DTM interval for our system setup.

Energy Consumption. As expected, DTM schemes for memory systems also affect their energy consumption. The energy consumption is related to the total memory traffic and running time. As discussed earlier, the DTM-ACG scheme is the most effective in reducing both the amount of memory traffic and the overall running time; it also reduces the memory energy consumption the most. Compared with the DTM-TS scheme, its average memory energy saving is 16.2% and 16.5% under the two configurations FDHS_1.0 and AOHS_1.5, respectively. Other DTM schemes only change the memory energy consumption slightly. Our experiments do not consider the use of DRAM low power mode because of the memory access intensity of the selected workloads.

Our proposed schemes, DTM-ACG and DTM-CDVFS, proactively control memory activities from the processor side. As a positive side effect, they can also reduce the energy consumption of the processor. We have used the data from Intel specification [13] to estimate their effects on the processor energy consumption. We plan to use power simulators such as Wattch [3] in the future to get more precise results. Compared with the DTM-TS scheme, our two proposed schemes reduce the processor energy consumption for every workload. The reason is that they scale down the number of active cores, the processor speed and voltage, and thus improve the processor energy efficiency, when the memory is to be overheated and the overall performance cannot benefit from additional processor computation power. Under the FDHS_1.0 configuration, compared with the DTM-TS scheme, the DTM-ACG and DTM-CDVFS schemes can reduce the processor energy consumption by 23.8% and 37.4% on average, respectively. By comparison, the conventional DTM-BW scheme, which throttles the memory bandwidth but may keep all processor cores running at full speed, consumes 45.1% more energy on average than the DTM-TS scheme, which shuts down both the processor and memory when the memory is to be overheated. This also emphasizes the importance to coordinate the control on the processor and on the memory.

6. RELATED WORK

Dynamic thermal management (DTM) has attracted increasing research interests in recent years. Most studies so far have focused on processors or disk drives. Brooks and Martonosi use the average processor power consumption over a time window as a proxy for chip temperature, and investigate several trigger, response and initiation mechanisms for DTM, such as scaling the clock frequency or voltage when the processor power consumption exceeds the preset trigger threshold [2]. Skadron et al. propose adaptive techniques based on control theory to manage the processor temperature and develop a thermal model for individual functional blocks [23]. They further extend the model to HotSpot, which models thermal behavior at microarchitecture level using a network of thermal resistances and capacitances, and can identify the hottest unit on chip [24]. They also propose several DTM techniques, such as migrating computation to spare hardware units from overheated ones. Li et al. studied the thermal constraints in the design space of CMPs [18]. To reduce simulation time, their simulator decouples processor core simulation and L2 cache simulation; and the overall L2 cache access trace is constructed from the L2 cache access traces of individual cores. In this way, the cache configuration can be varied without rerunning detailed simulation. Donald and Martonosi explore the design space of thermal management techniques for multicore processors [6]. Their study indicates that combining control-theoretic distributed DVFS and sensor-based migration policies performs the best. Another focus of research on thermal control is the hard disk drives. Gurumurthi et al. develop models to capture the capacity, performance and thermal behavior of disk drives. They also present two DTM techniques for hard disks, exploiting the thermal slack or throttling disk activities [9]. Kim et al. further present a performance-temperature simulator of disk drives and study the thermal characteristics and management of storage systems using server workloads [16].

Several studies have focused on reducing the power consumption of main memory systems. Although those proposed techniques may also help lowering the memory temperature, they do not directly target at alleviating the memory thermal emergency. Lebeck et al. propose a power-aware page allocation scheme that utilizes the long-latency but low-power DRAM modes. It minimizes the number of memory chips used by an application to increase the possibility that a DRAM chip can be put into low-power modes without affecting overall performance [17]. Delaluz et al. further propose using compiler techniques to map memory pages with similar active periods to the same chips in order to allow DRAM chips to stay in low-power modes longer [5]. Fan et al. study memory controller policies considering DRAM power states for power saving [7]. Huang et al. design and implement power-aware virtual memory management to save power consumption of main memory systems [12].

Our work focuses on dynamic thermal management of

memory systems, which is different from those studies. To our best knowledge, the work closest to ours is done by Iyer et al. [15]. Their work uses Delta Temperature in Serial Presence Detect and Thermal Sensor on a DIMM to control memory throttling for power and temperature management of Small Outline-DIMM (a DRAM module used in mobile platforms). One of their methods is close to the DTM-BW method that throttles memory bandwidth at the memory controller when the DRAM memories overheat. Our methods are different in that they throttle the processor core execution instead of simply limiting memory bandwidth. Since their work is for different platforms and different DRAM structures, the techniques cannot be directly applied to the systems targeted at in our study.

7. CONCLUSIONS

In this study, we have proposed two new DTM schemes, adaptive core gating (DTM-ACG) and coordinated DVFS (DTM-CDVFS) for DRAM thermal control. These schemes have the advantage of coordinating the current running states of the processor and the memory system. We have also built a two-level simulation infrastructure for evaluating the schemes on multicore processors with FB-DIMM memory. Our results show that both schemes can improve performance effectively for a given thermal limit, compared with existing memory thermal control techniques. In addition, they may also improve the processor energy efficiency. We have also applied the PID formal control theory method into the DTM-ACG and DTM-CDVFS schemes to further improve their effectiveness. This work is partially based on preliminary experiments on a real FB-DIMM system. In the future, we plan to evaluate the DTM schemes in full scale on real computer systems. We also plan to investigate more DTM schemes including the combination of DTM-ACG and DTM-CDVFS.

8. AKNOWLEDGEMENT

We appreciate the constructive comments from the anonymous reviewers. This work is supported in part by the National Science Foundation under grants CPA-0541408 and CPA-0541366.

9. REFERENCES

[1] N. L. Binkert, R. G. Dreslinski, L. R. Hsu, K. T. Lim, A. G. Saidi, and S. K. Reinhardt. The M5 simulator: Modeling networked systems. *IEEE Micro*, 26(4):52–60, 2006.

[2] D. Brooks and M. Martonosi. Dynamic thermal management for high-performance microprocessors. In *Proceedings of the 7th International Symposium on High-Performance Computer Architecture*, pages 171–182, 2001.

[3] D. Brooks, V. Tiwari, and M. Martonosi. Wattch: A framework for architectural-level power analysis and optimizations. In *Proceedings of the 27th International Symposium on Computer Architecture*, pages 83–94, 2000.

[4] D. C. Burger and T. M. Austin. The simplescalar tool set, version 2.0. Technical Report CS-TR-1997-1342, University of Wisconsin, Madison, 1997.

[5] V. Delaluz, M. T. Kandemir, N. Vijaykrishnan, A. Sivasubramaniam, and M. J. Irwin. DRAM energy management using software and hardware directed power mode control. In *Proceedins of the 7th International Symposium on High-Performance Computer Architecture*, pages 159–170, 2001.

[6] J. Donald and M. Martonosi. Techniques for multicore thermal management: Classification and new exploration. In *Proceedings of the 33rd International Symposium on Computer Architecture*, pages 78–88, 2006.

[7] X. Fan, C. Ellis, and A. Lebeck. Memory controller policies for DRAM power management. In *Proceedings of the 2001 International Symposium on Low Power Electronics and Design*, pages 129–134, 2001.

[8] B. Ganesh, A. Jaleel, D. Wang, and B. Jacob. Fully-buffered DIMM memory architectures: Understanding mechanisms, overheads and scaling. In *Proceedings of the 13th International Symposium on High Performance Computer Architecture*, 2007.

[9] S. Gurumurthi, A. Sivasubramaniam, and V. K. Natarajan. Disk drive roadmap from the thermal perspective: A case for dynamic thermal management. In *Proceedings of the 32nd International Symposium on Computer Architecture*, pages 38–49, 2005.

[10] J. Haas and P. Vogt. Fully-Buffered DIMM technology moves enterprise platforms to the next level. `http://www.intel.com/technology/magazine/computing/fully-buffered-dimm-0305.pdf`, 2005.

[11] A. R. Hambley. *Electrical engineering: Principles and applications*, pages 143–147. Prentice-Hall, Inc., 2nd edition, 2002.

[12] H. Huang, P. Pillai, and K. G. Shin. Design and implementation of power-aware virtual memory. In *USENIX Annual Technical Conference, General Track*, pages 57–70, 2003.

[13] Intel Corp. Dual-core Intel® Xeon® processor 5000 series. `ftp://download.intel.com/design/Xeon/datashts/31307901.pdf`, 2006.

[14] Intel Corp. Intel® fully buffered DIMM specification addendum. `http://www.intel.com/technology/memory/FBDIMM/spec/Intel_FBD_Spec_Addendum_rev_p9.pdf`, 2006.

[15] 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*, 10, 2006.

[16] Y. Kim, S. Gurumurthi, and A. Sivasubramaniam. Understanding the performance-temperature interactions in disk I/O of server workloads. In *Proceedings of the 12th International Symposium on High-Performance Computer Architecture*, pages 176–186, 2006.

[17] A. R. Lebeck, X. Fan, H. Zeng, and C. Ellis. Power aware page allocation. In *Proceedings of the 9th International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 105–116, 2000.

[18] Y. Li, B. Lee, D. Brooks, Z. Hu, and K. Skadron. CMP design space exploration subject to physical constraints. In *Proceedins of the 12th International Symposium on High-Performance Computer Architecture*, pages 71–82, 2006.

[19] D. Liaptan. FB-DIMM mechanical heat spreader design methdology, 2006. Intel Developer Forum.

[20] K. Man. Bensley FB-DIMM performance/thermal management, 2006. Intel Developer Forum.

[21] Micron Technology, Inc. DDR2 SDRAM system-power calculator. `http://www.micron.com/support/designsupport/tools/powercalc/powercalc`.

[22] T. Sherwood, E. Perelman, G. Hamerly, and B. Calder. Automatically characterizing large scale program behavior. In *Proceedings of the 10th International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 45–57, 2002.

[23] K. Skadron, T. Abdelzaher, and M. R. Stan. Control-theoretic techniques and thermal-RC modeling for accurate and localized dynamic thermal management. In *Proceedings of the 8th International Symposium on High-Performance Computer Architecture*, pages 17–28, 2002.

[24] K. Skadron, M. R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan. Temperature-aware microarchitecture. In *Proceedings of the 30th International Symposium on Computer Architecture*, pages 2–13, 2003.

[25] Standard Performance Evaluation Corporation. *SPEC CPU2000*. `http://www.spec.org`.

[26] D. C. Steere, A. Goel, J. Gruenberg, D. McNamee, C. Pu, and J. Walpole. A feedback-driven proportion allocator for real-rate scheduling. In *Operating Systems Design and Implementation*, pages 145–158, 1999.

[27] Q. Wu, P. Juang, M. Martonosi, and D. W. Clark. Formal online methods for voltage/frequency control in multiple clock domain microprocessors. In *Proceedings of the 11th International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 248–259, 2004.