Central vs. Distributed Dynamic Thermal Management for Multi-Core Processors: Which one is better?

Michael Kadin
Division of Engineering
Brown University
Providence, RI 02912

Sherief Reda
Division of Engineering
Brown University
Providence, RI 02912

Augustus Uht
Department of Electrical,
Computer and Biomedical
Engineering
University of Rhode Island
Kingston, RI 02881

ABSTRACT

In this paper we investigate and contrast two techniques to maximize the performance of multi-core processors under thermal constraints. The first technique is a distributed dynamic thermal management system that maximizes the total performance without exceeding given thermal constraints. In our scheme, each core adjusts its operating parameters, i.e., frequency and voltage, according to its temperature which is measured using integrated thermal sensors. We propose a novel controller that dynamically adapts the system to simultaneously avoid timing errors and thermal violations. For comparison purposes, we implement a second technique based on a runtime centralized, optimal system that uses combinatorial optimization techniques to calculate the optimal frequencies and voltages for the different cores to maximize the total throughput under thermal constraints. To empirically validate our techniques, we put together an extensive tool chain that incorporates thermal and power consumption simulators to characterize the performance of multi-core processors for a number of configurations ranging from 2 cores at 90 nm to 16 cores at 32 nm. Our results show that both investigated techniques are capable of delivering significant improvements (about 40% for 16 cores) over standard frequency and voltage planning techniques. From the results, we outline the main advantages and disadvantages of both techniques.

ACM Categories & Subject Descriptors
B.7.2 [Integrated Circuits]: Design Aids
General Terms: Design, Performance, Algorithms
Keywords: Thermal management, timing, DVFS.

1. INTRODUCTION

Elevated temperatures from increased on-chip power densities are a true limiter to scaling device speeds and circuit integration. The heat generated from dynamic and leakage power in state-of-the-art technologies (65 nm and below) prohibit the increase of operational clock frequency despite the potential of faster silicon. Heat generation is perhaps the greatest challenge to charge-based devices. Such a "thermal wall" could be alleviated by a number of techniques, including: (1) power reduction techniques; (2) better thermal management techniques; (3) better heat removal using active cooling devices; or ultimately (4) introduction of new, better device technologies. Besides being constrained by temperature, performance maximization has to adapt according to various workloads, as well as to process, operational, and environmental variations. By spatially and temporally adjusting the operating parameters, i.e., frequencies and voltages, the total performance can be adapatively maximized without violating the temperature constraints. Independent control of the operating parameters of individual cores is becoming feasible in state-of-the-art processors. For example, each core in AMD's Griffin family of processors has its own phase locked loop, clock distribution network and power grid, which allows independent control of the operating frequencies and voltages. In the Griffin family, frequency and voltage control is primarily driven by power and not temperature.

The overarching goal of this paper is to investigate and contrast two techniques to optimize the performance of multi-core processors under thermal constraints. The first technique is a proposed Distributed Dynamic Thermal Management (D^{2}TM) scheme, and the second technique is a runtime centralized, optimal frequency and voltage planning scheme. The contributions of this paper can be summarized as follows.

- We investigate a distributed dynamic thermal management technique that adaptively maximizes the performance of multi-core processors according to the workloads, process and environmental variations. We propose a novel feedback controller design that adjusts frequency and voltage such that no thermal or timing error violations can occur. We tune the controllers to maximize the processor's total performance under thermal constraints. Our work complements existing approaches (e.g., [3])) by incorporating a novel timing-error avoidance thermal controller. Our controller design enables it to simultaneously push the processor's speed as allowed by the fabricated silicon and without leading to thermal violations.
- Our work is the first to directly contrast distributed feedback-based DVFS against an optimization-based centralized mechanisms (proposed earlier by the authors [4]) that finds the best frequencies and voltages of all cores to maximize the processor's performance during runtime under thermal constraints.
- We put together a sophisticated tool chain using thermal simulators, power calculators, together with various workloads and processor configurations to validate the proposed ideas. We study four different processor configurations from 2 cores at 90 nm technology to 16 cores at 32 nm technology. Our results

GLSVLSI'09, May 10–12, 2009, Boston, Massachusetts, USA.
Copyright 2009 ACM 978-1-60558-522-2/09/05 ...$5.00.

show that the proposed techniques can deliver about 14% improvement for 4-core systems, 20% improvement for 8-core systems, and 40% improvement for 16-cores systems. The results of the D^{2}TM technique are contrasted to those of the centralized optimal technique and the advantages and disadvantages of both techniques are outlined.

The organization of this paper as follows. Section 2 introduces the main elements of our proposed distributed, dynamic thermal management technique. Section 3 provides extensive experimental results verifying the effectiveness of our proposed approach. Finally, Section 4 summarizes the main conclusions of this work.

2. PROPOSED APPROACH

The main constraint holding multi-core processors from performing better is that the maximum dynamic temperature anywhere on the chip must be kept below the chip's specified-maximum temperature, effectively throttling other circuits on the chip. Worst-case operating, environmental and manufacturing variations all combine to require a large safety margin to ensure correct operation even if all chip parameters are simultaneously at their worst-case values. Thus, the clock frequency is set to a value that ensures that no thermal violations or timing errors will occur anywhere on the chip, given any and all other worst-case conditions throughout the chip.

While using a single clock makes for a conceptually simple design, the resulting performance of the entire chip is much lower than necessary. The total effect of all of the worst-case-assumed timing margins is much lower performance than is needed under typical operating, environmental and process conditions. Furthermore, a fixed clock frequency results in a very poor heat distribution throughout the chip. Many cores do not operate near the chip's maximum temperature; thus, modulo increased power needs, these cores could run with a substantially higher clock frequency resulting in much better total system performance. In this section we investigate a *Distributed Dynamic Thermal Management System* (D^{2}TM) that aims to maximize the total performance under temperature constraints. We start by describing the main ingredient, the timing-error avoidance thermal controller, of our system.

2.1 TEA Thermal Controller

DVFS-based timing controllers aim to operate a system at or almost at a point of system timing failure or *timing error*. Timing Error Avoidance (TEA) schemes [8] employs a *tracking logic* to determine when an actual-logic error is about to occur, that is if the clock frequency increases further. The tracking logic is a one-bit wide copy of the worst-case delay path through the actual logic; the copy is faithful to the actual logic's components, wiring, and layout. Figure 1 gives an example of tracking logic in which the critical path of an adder (i.e., the carry propagate chain) is used for tracking with an additional delay for safety margin. At or soon after the “future" error is detected, the TEA controller reduces the clock frequency, increases the core voltage, or a combination of the two; the net effect is to increase the clock period. If no error is anticipated the clock frequency is increased. Since errors are avoided the only recovery necessary is to directly or indirectly decrease the clock frequency. Using TEA controllers requires no microarchitectural changes and incurs no loss in in cycles.

The backbone of our system is a timing-error avoidance thermal-driven DVFS based controller, which will refer to by just *TEA thermal controller*. The schematic of the controller is given in Figure 2, where each core has one of these controllers. The controller generally keeps the temperature of each core at about the maximum specification by varying the core's clock frequency, and optionally varying its core voltage as well. A core's temperature is compared to the set temperature (typically 85°C). The difference in temperature, i.e., *the error* $e(t)$, is sampled every Δ_s seconds using an analog-to-digital converter (represented by K_{TN}), and then the error is fed to a Proportional-Integral-Derivative (PID) process controller. The PID controller function is to produce a fast, stable system response. Options for the PID controller will be discussed in Section 2.2. The output from the controller is converted to frequency (K_{Nf}), and fed to both the core and the TEAvoidance circuitry. The TEAavoidance circuitry adjusts the frequency by incrementally lowering or raising the register 'N' value feeding the clock synthesizer. If the TEA circuit detects that a timing error is about to occur in the core, the $down/\overline{up}$ line is set to 1, lowering N and thereby reducing the clock frequency; otherwise, the $down/\overline{up}$ signal is cleared to 0, raising N and thereby increasing the frequency. Therefore the clock frequency of the core is approximately maximized for any value of the core's temperature. Note that the clock synthesizer may be realized with either a DAC/VCO combination, or by cycle skipping. A simple mapping function is used to optionally change the core voltage. In this case the core voltage is set in proportion to the square root of the clock frequency; a small lookup table can be used for this purpose in the actual hardware.

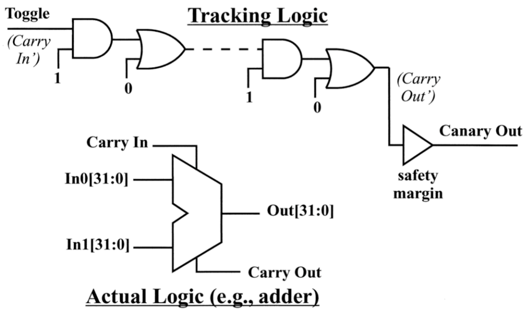

Figure 1: An example of the tracking logic in which the critical path of an adder (i.e., the carry propagate chain) is used for tracking with an additional delay for safety margin.

The MIMO (multi-input multi-output) multi-core chip has no centralized components; see Figure 3. Each core has its own clock synthesizer and optionally its own core voltage regulator. Control of the chip-system is fully distributed: every core has its own TEA thermal controller. The different controllers are only coupled by their own and their neighbors' heat generation. Each TEA thermal controller ensures that its core never exceeds the maximum-temperature specified for the chip. Further, the controller approximately maximizes its own core's performance by increasing the core's clock frequency as much as possible. Distributing the control has a couple of advantages. First, the chip layout and design is simplified. Second, the distributed control overhead scales linearly with the number of cores on a chip.

2.2 Controller Tuning and Stability

The PID controller in the proposed feedback TEA thermal controller (Figure 2) is crucial towards the stability of the system as it works on correcting the error between the measured temperature and the given temperature setting. A PID controller generally consists of three independent terms: the proportional term (P), the integral term (I), and the derivative term (D). The output from three terms is $K_P e(t) + K_I \int_0^t e(m)dm + K_D \frac{de(t)}{dt}$. The *proportional term* produces an output that is proportional to the error value $e(t)$. A high proportional gain K_P can lead to an oscillatory behavior, and in the absence of disturbances, a pure proportional control will not settle at its steady-state target value. The *integral term* is propor-

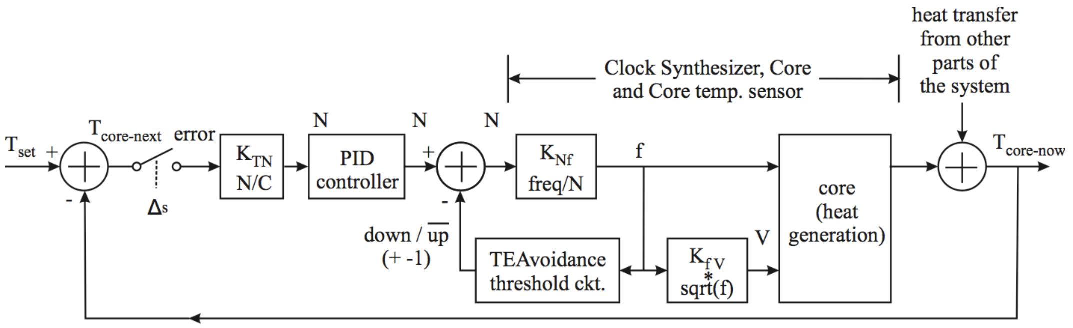

Figure 2: Per-core Timing-error avoidance Thermal DVFS controller. Δ_s is the sample period. Using a lookup table, the frequency is sent to 'sqrt(f)' to link the core voltage to the core's clock frequency.

tional to both the magnitude of the error and the duration of the error. The integral controller accumulates the error and then multiply it by the integral controller gain K_I. When added to the proportional term, integral term accelerates the movement of the system towards the set point and eliminates the steady-state residual error that occurs in the proportional term. However, an overshoot above the set point value might occur as the controller responds to accumulated errors. The amount of overshoot can be reduced by using a *derivate term* that slows down the rate of change of controller output when it is integrated with PI or I controllers. However, differentiation of the input signal can lead to detrimental effects if substantial noise is present in the input signal. In our work we only use PI controllers and manually tune the gain of the system by carefully adjusting the gain to avoid overshoot and over-damped response. In-depth investigation of MIMO system stability is beyond the limited space of this paper; however, the specific values used in the experiments do provide a stable system.

3. EXPERIMENTAL RESULTS

To emulate the impact of our methodology during runtime, we set up a tool chain or a simulation infrastructure to determine the impact of changing the frequencies, voltages and workloads on the temperatures of various processor units during runtime. Our tool chain takes as inputs: the individual cores operating parameters, the specification of the processor circuitry, and the workload tasks that the system will run on the cores. Using these input parameters, the dynamic power of each unit is calculated and then fed into a thermal simulator. Once the temperatures have been calculated, they are fed into the leakage calculator to compute the leakage of each unit. The leakage power is then added to the original dynamic power values, and new temperatures are calculated. The process is iterated until the temperatures converge. The process usually takes a few iterations (typically < 10) to reach stable convergence. To compute the power and temperature values, we use the following established tools.

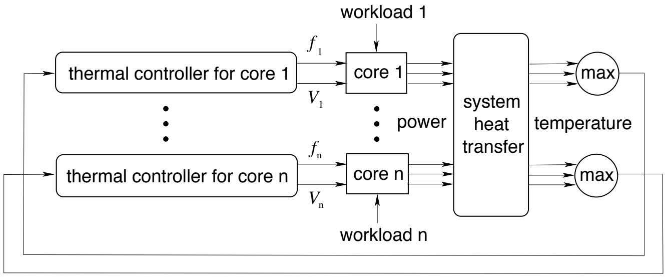

Figure 3: Modeling of a multi-core system as a Multiple Input Multiple Output System.

- For dynamic power simulation we use PTScalar [6]. PTScalar is a Wattch-like [1] power simulator built as an extension of the of Simple-Scalar toolset [2].
- We also take the expressions for leakage power from PTScalar, though we construct our own leakage power calculator in order to maintain tool versatility. To accurately model cache leakage power, we use CACTI 5.0 [9], which has accurate cache leakage values at current and coming technology nodes.
- We utilize HotSpot (version 4.0) [7] as our temperature simulator.

We use the Alpha EV6 processor as our baseline and scale it appropriately for each technology generation, leading to four processor configurations: 2 cores at 90 nm, 4 cores at 65 nm, 8 cores at 45 nm, and 16 cores at 32 nm. In all configurations the total die area is kept at 124.01 mm^2 organized as 11.13 mm × 11.13 mm, where the L2 cache occupies 85.25 mm^2 and the cores occupy 38.76 mm^2. A representative subset of the SPEC2000 bencmarks is used in the experiments: four integer workloads (bzip, gcc, twolf, and mcf) and four floating-point workloads (ammp, equake, lucas, and mesa). In all experiments, we set the sampling interval Δ_s to 0.3 seconds. Thus the D^{2}TM controllers are engaged every 0.3 seconds to measure the temperatures of the cores and adjust the frequencies and voltages accordingly. We set the maximum specification core speed to 3.5 GHz. We only use PI controllers for the D^{2}TM system, and after tuning the feedback controller, we found that a controller gain of 0.06 gives a stable response.

In our experiment we directly contrast the performance of runtime optimal, central thermal management against distributed thermal management. For the runtime centralized approach, we implement the frequency and voltage planning technique proposed earlier by the authors [5, 4]. We used MATLAB to solve the runtime centrail optimization methods we proposed. MATLAB runtime is usually between $0.5 - 1.5$ seconds. It is expected that a lean C-based implementation could be $10 - 100\times$ faster than MATLAB.

As the performance under thermal constraints vary depending on the thermal aggressiveness of the workloads, we contrast the performance of the two techniques using various workload configurations. Table 1 gives four workload configurations that present a good mixture of typical multi-core processor workloads. Each configuration has 16 workloads. Configuration I is a mix of eight integer and floating point workloads; configuration II is a mix of four integer workloads; configuration III is a mix of four floating point workloads; and configuration IV is a mix of relatively thermal-aggressive workloads. In Table 1 we report 16 workloads for each configuration. Workloads are assigned to cores in the given order. For example, in configuration I, core 1 is assigned bzip, core 2 is assigned gcc, and so forth. If the processor has less than 16

Configuration	Workloads
I	bzip, gcc , twolf, mcf, ammp, equake, lucas, mesa, bzip, gcc , twolf, mcf, ammp, equake, lucas, mesa
II	bzip, gcc , twolf, mcf, bzip, gcc , twolf, mcf, bzip, gcc , twolf, mcf, bzip, gcc , twolf, mcf, bzip, gcc , twolf, mcf
III	ammp, equake, lucas, mesa, ammp, equake, lucas, mesa, ammp, equake, lucas, mesa, ammp, equake, lucas, mesa
IV	gcc, equake, bzip, mesa, gcc, equake, bzip, mesa, gcc, equake, bzip, mesa, gcc, equake, bzip, mesa

Table 1: Workload configurations. Workloads are assigned to cores in order.

Configuration	Number of cores = 2			Number of cores = 4			Number of cores = 8			Number of cores = 16		
	STAND	OPT	D^2TM	STAND	OPT	D^2TM	STAND	OPT	D^2TM	STAND	OPT	D^2TM
Workload configuration I	6.26	6.32	6.31	12.08	13.81	13.72	22.30	26.44	26.21	34.63	49.02	48.33
Workload configuration II	6.26	6.32	6.31	12.08	13.81	13.72	21.33	26.01	25.82	33.56	48.42	47.79
Workload configuration III	6.68	6.80	6.79	12.78	14.00	14.00	22.22	26.62	26.41	34.91	49.63	48.89
Workload configuration IV	6.34	6.38	6.37	11.53	13.64	13.58	20.47	25.73	25.52	34.38	47.79	47.15
Average	6.38	6.46	6.45	12.12	13.82	13.76	21.58	26.20	25.99	34.37	48.72	48.04
Adv. over STAND (%)	0.00%	1.10%	0.94%	0.00%	14.01%	13.51%	0.00%	21.41%	20.44%	0.00%	41.74%	39.77%

Table 2: Total processor throughput (as measured by the sum of all core frequencies) for different management techniques.

	Opt. Central	D^2TM
Advantages	• Maximum performance guaranteed • Flexible to incorporate any additional constraints • Scalable for many-core systems	• Near-maximium performance • Copes better with manufacturing variations in fabricated silicon • Number of controllers scales linearly with the number of cores
Disadvantages	• SW Optimization could add to processor workload • Modeling inaccuracies could override temperature constraint	• Must be well-tuned or else system could be unstable • System could take a while to reach the highest performance • HW overhead to implement feedback controllers and TEA circuitry

Table 3: Comparison of various thermo-performance control using various processor configuration.

cores then it is assigned in order a number of workloads equal to its number of cores.

In Table 2 we report the total throughput results (as measured by the sum of all frequencies) of three techniques: (1) standard planning, (2) D^2TM, and (3) optimal, central frequency and voltage planning [5, 4]. Standard planning is the processor's throughput when all cores are run at exactly the same frequency. We report the sum of all core frequencies (in GHz) for various processor configuration and workload configurations. From the results, it is clear that both optimal planning and distributed management deliver superior performance (e.g., up to 40% better performance in the 16-core case) in comparison to standard planning. Furthermore, the magnitude of performance improvement increases as the number of cores increases. We also note that optimal planning slightly outperforms D^2TM. While it is not hard to expect that optimal planning fares better than distributed management, the small magnitude of improvement demonstrates that D^2TM is a viable method for thermal management. In Table 3 we outline that main advantages and disadvantages of both runtime optimal planning and distributed management. Optimal planning could be advantageous when some constraints are required to be imposed on the cores. For example, it could be desirable to equate the frequencies of a number of cores to guarantee fairness or to minimize the maximum time a workload needs for execution. Optimal planning also uses software (SW) as its main vehicle to model the processor's behavior and to find its optimized operating parameters. Using SW-based thermal management systems is advantageous as it reduces the hardware (HW) complexity; however, it could be a bottleneck in case the runtime of model optimization overloads the processor. Dynamic D^2TM methods could be better poised to cope with manufactured variability; for example, our proposed hybrid timing-thermal controller is capable of pushing the processor's performance without violating both the processor's timing and temperature specifications.

4. CONCLUSIONS

We have investigated a new distributed dynamic thermal management method to maximize the performance of multi-core processors under thermal constraints. We have provided the design of a timing-error avoidance thermal controller, and explained how the entire processor operates like automatic process control systems. We have leveraged this relationship to design controllers that provide stable dynamic response. We have also analyzed the performance of the system in comparison to a runtime central thermal management system that uses a combinatorial optimization framework to arrive to the optimal frequencies and voltages of the processor. To test the proposed approach, we put forward a comprehensive tool chain of thermal and power simulators, and tested the tool chain on various processor configurations with different workloads. Our results maximize the performance of multi-core systems (by an average of 40% for 16-core processors) without exceeding the temperature specifications.

5. REFERENCES

[1] D. Brooks, V. Tiwari, and M. Martonosi, "Wattch: A Framework for Architectural-Level Power Analysis and Optimizations," in *International Symposium on Computer Architecture*, 2000, pp. 83–94.

[2] D. C. Burger and T. M. Austin, "The SimpleScalar Tool Set, Version 2.0, Tech. Rep. CS-TR-1997-1342, 1997. [Online]. Available: citeseer.ist.psu.edu/burger97simplescalar.html

[3] J. Donald and M. Martonosi, " Techniques for Multicore Thermal Management: Classification and New Exploration," in *International Symposium on Computer Architecture*, 2006, pp. 78–88.

[4] M. Kadin and S. Reda, "Frequency and Voltage Planning for Multi-Core Processors Under Thermal Constraints," in *International Conference on Computer Design*, 2008, pp. 463–470.

[5] M. Kadin and S. Reda, "Frequency Planning for Multi-Core Processors Under Thermal Constraints," in *International Symposium on on Low Power Electronics and Design*, 2008, pp. 213–216.

[6] W. Liao, L. He, and K. Lepak, "Temperature and Supply Voltage Aware Performance and Power Modeling at Microarchitecture Level," *Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24(7), pp. 1042–1053, 2005.

[7] K. Skadron, S. Ghosh, S. Velusamy, K. Sankaranarayanan, and M. Stan, "HotSpot: A Compact Thermal Modeling Methodology for Early-Stage VLSI Design," *Transactions on VLSI Systems*, vol. 15(5), pp. 501–513, 2006.

[8] A. K. Uht, "Going Beyond Worst-Case Specs with TEAtime," *Computer*, vol. 37, no. 3, pp. 51–56, 2004.

[9] S. Wilton and N. P. Jouppi, "CACTI: An Enhanced Cache Access and Cycle Time Model," *IEEE Journal Solid-State Circuits*, vol. 31(5), pp. 677–688, 1996.