

Impact of Thermal Constraints on Multi-Core Architectures

Yingmin Li[†], Benjamin Lee[‡], David Brooks[‡], Zhigang Hu^{††}, Kevin Skadron[†]

[†] Dept. of Computer Science, University of Virginia ^{††} IBM T.J. Watson Research Center

[‡] Division of Engineering and Applied Sciences, Harvard University

{yingmin,skadron}@cs.virginia.edu, zhigangh@us.ibm.com, {dbrooks,bclee}@eecs.harvard.edu

ABSTRACT

This paper shows how thermal constraints affect the multi-dimensional design space for chip multiprocessors, considering the inter-related variables of CPU count, pipeline depth, superscalar width, L2 cache size, and operating voltage and frequency. The results show the importance of thermal modeling and the need for new thermal modeling capabilities and hence the need for collaboration between the thermal engineering and computer architecture communities. Thermal constraints both shift the optimal intra- and inter-core organization, and dominate other physical constraints such as pin-bandwidth and power delivery. Different thermal constraints also require different optimization strategies. For aggressive cooling solutions, reducing power density is at least as important as reducing total power, while for low-cost cooling solutions, reducing total power is more important.

KEYWORDS: multi-core architecture, thermal management, compact thermal models

NOMENCLATURE

FO4: Fan-out-of-four delay—delay of one inverter driving four copies of an equally sized inverter. Deeper pipelines have smaller FO4 delays.

pipeline depth: number of stages of processing required for each instruction

pipeline width: number of instructions that can begin execution in parallel

L1 cache: primary cache for a core—can usually be accessed within a few cycles, with sizes of 8–128KB

L2 cache: secondary cache, sometimes shared among multiple cores, access times of 10–20 cycles, with sizes of 256KB up to several megabytes.

INTRODUCTION

Recent product announcements show a trend toward aggressive integration of multiple CPUs (“cores”) on a single chip to maximize throughput. However, this trend presents an expansive design space for chip architects, encompassing the number of cores per die, core size and complexity (pipeline depth and superscalar width), memory hierarchy design, operating voltage and frequency, and so forth. Identifying optimal designs is especially difficult because the variables of interest are inter-related and must be considered simultaneously. Trade-offs also vary depending both on workloads and physical (e.g., area and thermal) constraints.

This paper summarizes our previous work [1] in order to show how thermal modeling capabilities can drive computer architecture and vice versa. The advent of large-scale multi-core chips is incompatible with traditional computer architecture simulation techniques and requires new performance as well as thermal modeling capabilities. The primary goal of the paper is to illustrate the need for collaboration between thermal engineers and computer architects.

Little prior work has considered so many cores. To our knowledge, this is the first work to optimize across so many design variables simultaneously and the first to propose a thermal modeling approach for multi-core architecture studies. This paper shows the inter-related nature of these parameters and how the optimum choice of design parameters can shift dramatically depending on system constraints. A design must be optimized with thermal constraints, which contravenes common practice. For example, core type is often optimized separately to maximize single-thread performance, core count optimized next to meet area and off-chip bandwidth constraints, and thermal design considered last. This approach will not be viable in future technology generations where large numbers of cores can be integrated on a single chip: scaling from the thermal-blind optimum leads to a configuration that is inferior, sometimes radically so, to a thermally optimized configuration. All these design decisions must be optimized jointly, but the ability to optimize across such a large design space requires a simple, fast approach to simulate a large number of cores by observing that cores only interact through the L2 cache and shared interconnect. The proposed methodology uses single-core traces and only requires fast cache simulation for multi-core results. This, however, requires a new dynamic compact thermal modeling approach that, as far as we know, differs from previous strategies and presents interesting precision/speed tradeoffs as well as numerous opportunities for improvement.

With the resulting infrastructure, the paper shows that simpler, smaller cores that otherwise would not be optimal are often preferred when thermal constraints are considered. In thermally constrained designs, the main determinant is not simply maximizing the number of cores, but maximizing their power efficiency. In particular, thermal constraints generally favor shallower pipelines and hence lower clock frequencies: deeper pipelines require more latches between processing stages, and because each stage does less work, deeper pipelines also permit higher clock frequencies. Taken together, these factors dramatically increase power density. Thermal constraints may also favor narrower pipelines. Many CPUs today can initi-

ate execution on multiple instructions in parallel; this “superscalar” execution raises power density in proportion to the superscalar execution width. A final consideration is that operating voltage and frequency can be scaled back from what a core can nominally sustain. Additional cores increase throughput even when voltage and frequency scaling are required to meet thermal constraints, until performance gains from an additional core is negated by the impact of voltage and frequency scaling across all cores. But voltage and frequency scaling also interact with pipeline depth and width, because a deep or wide core that is scaled back aggressively is inferior to a simpler core with lower power density that can operate at a higher frequency.

The nature of the thermal packaging also changes the trade-offs. For aggressive cooling solutions, reducing power density is at least as important as reducing total power. For low-cost cooling solutions, however, reducing total power is more important. It also turns out that, in both cases, thermal optimization necessitates reductions in voltage and frequency that reduce power enough so that power-delivery limits are also met. For the workloads studied here, these reductions in voltage and frequency slow down the cores enough that their off-chip access rate is also within projected pin-bandwidth capabilities.

RELATED WORK

There has been a burst of work in recent years to understand the performance, energy, and thermal efficiency of different multi-core organizations. Few have looked at a large numbers of cores, and none of which we are aware have jointly optimized across the large number of design parameters we consider nor addressed the associated methodology challenges. Huh et al. [2] categorized the SPEC benchmarks into CPU-bound, cache-sensitive, or bandwidth-limited groups and explored core complexity, area efficiency, and pin bandwidth limitations, concluding due to pin-bandwidth limitations that a smaller number of high-performance cores maximizes throughput. Ekman and Stenström [3] use SPLASH benchmarks to explore a similar design space for energy-efficiency with the same conclusions.

Kongetira et al. [4] describe the Sun Niagara processor, an eight-core chip supporting four threads per core and targeting workloads with high degrees of thread-level parallelism. Chaudhry et al. [5] describe the benefits of multiple cores and multiple threads, sharing eight cores with a single L2 cache. They also describe the Sun Rock processor’s “scouting” mechanism that uses a helper thread to prefetch instructions and data.

Other researchers have proposed simplified architectural processor models with the goal of accelerating simulation, but only for performance, e.g. in terms of instructions per cycle (IPC). A brief survey of these techniques can be found in [1].

EXPERIMENTAL METHODOLOGY

This study addresses early-stage architecture planning, in which the basic organization of a chip is set before detailed implementation begins. This design phase has huge leverage over

the final performance, energy-efficiency, and thermal properties of a product and dictates much of the implementation. These early architecture studies typically use cycle-by-cycle pipeline simulations with performance and power characteristics extrapolated from prior products.

Traditional, detailed, cycle-accurate simulations of multi-core organizations are expensive, and the multi-dimensional search of the design space, even with just homogeneous cores, is prohibitive. To facilitate the exploration of large multi-core design spaces, it is proposed that core and interconnect/cache simulation can be decoupled to reduce simulation time. Decoupling core and interconnect/cache simulation dramatically reduces simulation cost with minimal loss in accuracy. This approach was first described in [1].

Simulator Infrastructure

The performance and power modeling techniques used in this study are based on IBM’s Turandot/PowerTimer tools. This infrastructure is used to generate single-core L2 cache-access traces that are annotated with timestamps and power values. These traces are then fed to a new multi-core simulator we developed in conjunction with IBM’s MET/Turandot tools.¹ This simulator consists primarily of a cache simulator that models the interaction of multiple threads on one or more shared interconnects and one or more L2 caches. It uses hits and misses to shift the time and power values in the original traces. In other words, this approach separates characterization of individual cores from characterization of multi-core chips. A detailed core simulation provides performance and power data for various core designs, while interconnect/cache simulation projects the impact of core integration and interaction on these metrics. Generating the traces is therefore a one-time cost, while what would otherwise be a costly multiprocessor simulation is reduced to a much faster cache simulation. Using the new multi-core simulator, it is cost-effective to search the entire multi-core design space.

Core Simulation

Turandot [6] models an IBM POWER4-like architecture. PowerTimer [7] implements circuit-extracted power models, which were extended with analytical scaling formulas based on Wattch [8], a microarchitectural power model developed in academia. Each of these components is modular so that any particular simulator can be replaced with an alternative. Turandot and PowerTimer were also extended to model performance and power as pipeline depth and width vary, using techniques from prior work [1, 9].

Scaling for Pipeline Depth: Pipeline depth is quantified in terms of FO4 delays per pipeline stage. Making a pipeline deeper means dividing the work to process an instruction into smaller steps. Performance is determined chiefly by clock frequency, and this scales roughly linearly with pipeline depth, because each pipeline stage requires one clock cycle and smaller steps allow shorter clock cycles. In other words,

¹This multi-core simulator, which we call “Zauber”, has been incorporated with the MET/Turandot tools and is also available as open-source software to the public upon request.

the basic amount of work and time to process an instruction stays roughly the same, so deeper pipelines simply allow more overlap between successive operations, and this translates into higher clock frequencies. This is consistent with prior work in pipeline depth simulation [10]. Power also scales linearly with pipeline depth: $P \propto V \cdot f^2$, where V is voltage and f is frequency, and frequency is proportional to pipeline depth. A core's area scales with pipeline depth roughly linearly, except that the number of latches between pipeline stages grows superlinearly [10]. A more detailed explanation of how performance, power, and area scale with pipeline depth can be found in [1].

Scaling for Pipeline Width: Pipeline width is defined as the number of instructions in a single core that can initiate execution simultaneously. (Obviously, these instructions must be independent.) Performance is roughly proportional to width, but the full benefit of wider pipelines often cannot be realized because there are not enough independent instructions to occupy the full issue width. Power and area are also roughly proportional to width. A more detailed explanation of how performance, power, and area scale with pipeline width can be found in [1].

Interconnection/Cache Simulation

The proposed approach for simulating multi-core chips decouples detailed core simulation and the simulation of core interaction. The cores in a multi-core architecture usually share one or more L2 caches through an interconnection fabric. Therefore, resource contention between cores occurs primarily in these two resources. It is therefore possible to simulate cache and fabric contention independent of core simulations without losing too much accuracy. The impact of contention on the performance and power of each core may then be evaluated quickly using interpolation.

First, L2 cache-access traces must be generated for each benchmark, based on L1 cache misses through one pass of single-core simulations. These traces record the L2 cache address and access time (denoted by the cycle) information for every access. These traces are annotated with performance and microarchitectural resource utilization every 10k instructions, as this information will be used in the interpolation. These L2 traces are fed into a cache simulator and interconnection-contention model that reads the L2 accesses of each core from the traces, sorts them according to time of access, and uses them to drive the interconnection and L2 cache simulation. This interconnection/cache simulator outputs the L2 miss ratio and the delay due to contention for every 10k instruction segment of the thread running on each core.

With this L2 miss ratio and interconnection contention information, the new performance and power numbers for each 10k instruction segment of all the threads can be calculated. Since the performance and microarchitectural resource utilization are known for several L2 miss ratio values, new performance and utilization data can be derived for any other L2 miss ratio produced by the cache simulator via interpolation. Power numbers can be derived from the structure utilization data with post-processing.

When interleaving the L2 accesses from each thread, cycle information attached with each access is used to sort them by time of access. However, each thread may suffer different degrees of performance degradation due to interconnection and L2 cache contention. Therefore, sorting by time of access may not reflect the real ordering. Simply iterating improves accuracy. In particular, given the performance impact from cache contention for each thread, this information can be used to adjust the time of each L2 access in each trace and redo cache emulation based on this new L2 access timing information. Three iterations are typically sufficient to reach convergence.

Validating the above approach against a full, multi-core simulation using the detailed, cycle-accurate simulator shows that the simplified approach is both accurate and fast. Figure 1 shows the average performance and power differences for 2- and 4-core chips. The average performance and power difference is within 1%. For a 2-core chip, the proposed approach achieves a simulation time speedup of 40-60x, with detailed Turandot simulations requiring 1-2 hours and the decoupled simulator requiring 1-3 minutes.

Since throughput-oriented workloads consisting of independent threads are the focus of this study, a relatively high degree of cache sharing, like Sun's Niagara/T1 chip [4] is chosen. Each L2 cache bank is shared by half the total number of cores. The interconnection power overheads are extrapolated from [11]. Based on a POWER5 die photo, the baseline core area is estimated as $11.52mm^2$, equivalent to the area of 1MB of L2 cache. This study assumes each $n/2$ cores share one L2 cache through a crossbar routing over the L2 and estimate the total crossbar area to be $6.25n \cdot mm^2$, where n is the number of cores.

To simplify the study, it is assumed that the L2 cache latency does not change when the L2 cache size varies. The effects of clock propagation on chip throughput and power as core number increases has also been omitted.

Multi-core Thermal and Leakage Simulation

To accommodate the more abstract simulation proposed above, it is necessary to estimate temperature at a matching level of abstraction. This study uses workloads in which all cores are occupied with active threads, to create "worst typical-case" workloads that are likely to dictate thermal design. Steady-state power at the granularity of each core is used to estimate steady-state multi-core chip thermal effects.

Even with the accelerated simulation approach (which takes only 1-3 minutes per configuration), a high-level microarchitectural model such as HotSpot [12] was too slow for large design-space searches. A simple, analytic formula was needed, able to resolve temperature with at least a core granularity without significantly increasing the time of each of these fast simulations.

It seems likely that thermal constraints would primarily affect multi-core architectural decisions in terms of how many cores could be integrated and how large or aggressive those cores would be. Modeling power and temperature only at the granularity of the core and not the individual substructures on the chip (which have widely varying power densities

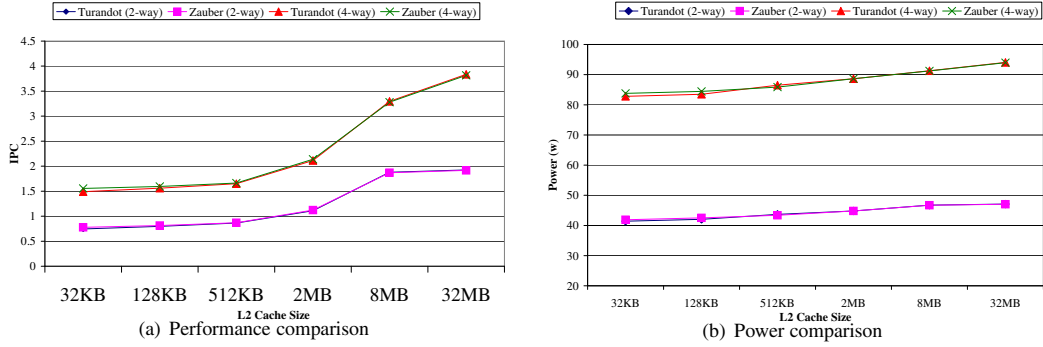


Figure 1. Validation of the proposed, decoupled abstract multi-core model [1].

and activity factors) neglects localized hotspots within a core. But temperature variations *within a core* can be dealt with by optimizing or redesigning those structures that are hotspots. Modeling power and temperature at the granularity of the core also neglects lateral thermal coupling among cores. Validation against HotSpot (see below) suggests that this is not a major source of error.

To develop an analytical expression, it is assumed that the heat spreader is almost isothermal for the range of the chip areas and power values investigated, so the global temperature rise across the thickness of the thermal package/heat sink due to total chip power dissipation can be separated from localized temperature rise above the package (i.e., above the heat spreader) due to per-core power dissipation. This is described by Equations 1–3. To calculate the temperature of a specific core on the chip, let P_{glo} and P_{core} be the global chip and single core dynamic power, respectively. Similarly, let L_{glo} and L_{core} be the global chip and single core leakage power, respectively. The chip’s total dynamic power is the sum of the dynamic power dissipated by all the cores on chip, the L2 cache and the interconnect. The chip leakage power is summed in a similar manner. The sum of R_{spread} and R_{hsink} denotes the thermal resistance from the heat spreader to the air and the sum of $R_{silicon}$ and R_{TIM} denotes the thermal resistance from the core. Collectively, these parameters specify the chip’s thermal characteristics from the device level to the heat spreader, ignoring the lateral thermal coupling above the heat spreader level.

Multi-core heatup can thus be categorized into local and global effects. The former is determined by the local power dissipation of any given core and the effect on its temperature. The latter is determined by the global chip power.

$$H_{glo} + H_{loc} = T_{core} - T_{amb} \quad (1)$$

$$H_{glo} = (P_{glo} + L_{glo}) \cdot (R_{spread} + R_{hsink}) \quad (2)$$

$$H_{loc} = (P_{core} + L_{core}) \cdot (R_{silicon} + R_{TIM}) \quad (3)$$

This distinction between local and global heatup mechanisms is first qualitatively introduced by Li et al. in [13]. This observation may not be evident strictly from the perspective of per-core power density. Although power density is often used as a proxy for steady-state temperature, with each core exhibiting the same power density, localized power density is only an accurate predictor of the temperature increases in the silicon

relative to the package. Per-unit or per-core power density is analogous to one of the many thermal resistances comprising the entire network that represents the chip.

For a given thermal package, adding cores increases temperature, because it increases the total amount of power that must be removed. Of course, the package itself is a design variable. The architecture field currently lacks a way to scale the thermal package in an analytical way in accordance with chip configuration. This is especially difficult because packaging for high-performance chips is already severely constrained. For example, fan speed may be increased, but this approach is often limited by acoustical limits and various board-layout and airflow factors that lead to diminishing returns (e.g. increased pressure drop across a larger heat sink). The inlet air temperature can be lowered, but this is not an option in many operating environments (e.g. a home office), or may be extremely costly (e.g. in a large data center). The heat spreader and heat sink size could also be increased, but high-performance packaging is already so large compared to the chip that this may not make a significant difference. All these considerations, in turn, are limited by form factor, airflow, and acoustic constraints that may be dictated by product or ergonomic goals, such as in laptops. Developing ways to model these tradeoffs in a way that is useful for early-stage architectural studies is an important area for future work. For these reasons, this paper simply explores the design space for fixed package thermal resistances.

Figure 2 presents results from validating this simple model against HotSpot 2.0 [12], an architectural model of localized, on-chip temperatures which has been validated against a thermal test chip [14], an FPGA [15], and high-resolution finite-element simulations of a major commercial microprocessor. Figure 2 varies heat sink resistance for a representative multi-core configuration and compares the proposed analytical approach to HotSpot. The temperature difference between these two models is within 3° . It is important to note, however, that in this experiment, HotSpot is only set up to model each core as a monolithic block; in other words, there is no sub-structure modeled within the cores, and each core has a possibly unique but uniform power density. This result does, however, suggest that omission of local, lateral thermal coupling among the cores is acceptable.

Clearly, this simple analytical thermal model is only a first step. But it allows early exploration of how thermal constraints

interact with the multi-core design space. The goal of this paper is to use these results to illustrate the importance of thermal modeling in early-stage architecture studies and the requirements of such a model, with the hope that this will stimulate new, multi-disciplinary research between the thermal and architecture fields. Improvements in the above will be immensely valuable in early-stage planning studies of large design spaces such as presented by large, multi-core chips.

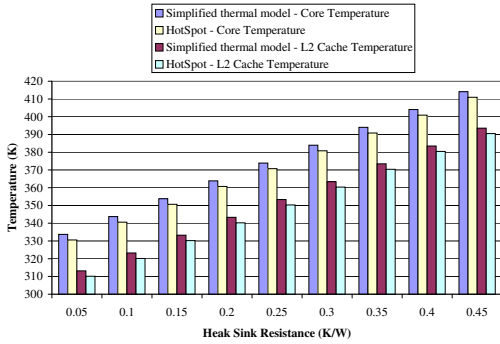


Figure 2. Simplified temperature model validation [1].

Accurate power and thermal modeling also requires accounting for leakage power—power that is dissipated even when transistors are not switching. This is especially important because its magnitude is exponentially dependent on temperature. This can be estimated according to Eq. (4), where A and B are coefficients determined by a linear regression of data from the International Technology Roadmap for Semiconductors and T is the absolute temperature. $A = 207.94$ and $B = 1446$ for 65nm technology.

$$P_{\text{leakage density}} = A \cdot T^2 \cdot e^{-B/T} \quad (4)$$

Voltage and Frequency Scaling for Thermal Control

Using a large number of cores may lead to thermal runaway due to high chip power and the positive feedback of leakage power and temperature. A thermal control mechanism is needed to prevent this behavior and to account for the resulting performance impact. With nominal operating parameters of 0.9 volts and 2.0 GHz, worst-case voltage and frequency are determined so that the peak, steady-state power dissipation observed for any workload will not push the steady-state temperature above 100°. Note that frequency is a hyperbolic function of voltage, with actual values derived from circuit simulations; in other words, for small changes in voltage, the resulting change in frequency is roughly proportional, but for larger changes in voltage, the required reduction in frequency is even larger.

Workloads

As benchmarks, this study employs eight, single-threaded SPECcpu2000 benchmarks (art, mcf, applu, crafty, gcc, eon, mgrid, swim). These are selected because they represent a range of instruction-level parallelism, memory requirements, and thermal characteristics. In particular, L2 miss rate turns out to be important, with high-miss-rate applications (i.e., memory-bound)—art and mcf—benefiting from substantially different configurations than low-miss-rate (i.e., CPU-bound) applications—the other six benchmarks.

For both CPU-bound and memory-bound benchmarks, simulations use pairs of single-thread benchmarks to form dual-thread benchmark groups and replicate these pairs to form multiple-benchmark groups of each benchmark category for multi-core simulation with more than two cores. Only a large pool of waiting threads is considered in order to keep all cores active, representing the “worst typical-case” operation likely to determine physical limits.

RESULTS

This design space exploration optimizes for performance (billions of instructions per second or BIPS) under various thermal constraints for a 400mm² chip—chosen because it is approximately similar in size to today’s POWER5 dual-core chip. In addition to demonstrating the effectiveness of the proposed experimental methodology for exploring large design spaces, the results also quantify significant multi-core design trends and demonstrate the need to make balanced design choices. Note that optimizing purely for aggregate throughput is suitable for servers, but allows the choice of simpler cores that might penalize single-thread execution latency. This will remain an important concern for personal computing devices, and balancing throughput and single-thread latency is an area for future work.

As mentioned earlier, in none of the experiments did a configuration exceed 250 W peak power draw or 30 GB/sec. Whether these observations would hold for other workloads is unclear, but they do suggest that designing for thermal limits can help meet other physical constraints such as power delivery and pin bandwidth.

Optimal Configurations

Figures 3–5 present performance trade-offs between core count, L2 cache size, and pipeline dimensions for a 400mm² chip subject to various thermal constraints. Packaging assumptions and hence thermal constraints can take on one of three values: no constraint (NT), low constraint (LR=0.1, low thermal resistance, i.e. aggressive, high-cost thermal solution), and high constraint (HR=0.5, high thermal resistance, i.e. constrained thermal solution, such as found in a laptop). A complete set of tables itemizing optimal configurations and their optimal voltage and frequency settings for 400mm² chips as well as unconstrained, 200mm², and 100mm² chips can be found in [1].

Thermal constraints tend to shift optimal configurations to fewer and simpler cores and less L2 cache. This is most clearly observed with the memory-bound benchmarks, which can benefit from up to 16 somewhat aggressively pipelined cores with no thermal limits but peak at 10 cores with shallow pipelines for the most constrained (R=0.5) heat sink.

Figure 5 also illustrates the impact of global heating on optimal pipeline configurations. As the number of cores increase for CPU-bound benchmarks, the optimal delay per stage increases by 6FO4 (i.e., from 18 to 24FO4—making the pipeline shallower) when twelve cores reside on a single chip. The increasing core count increases chip temperature, leading to

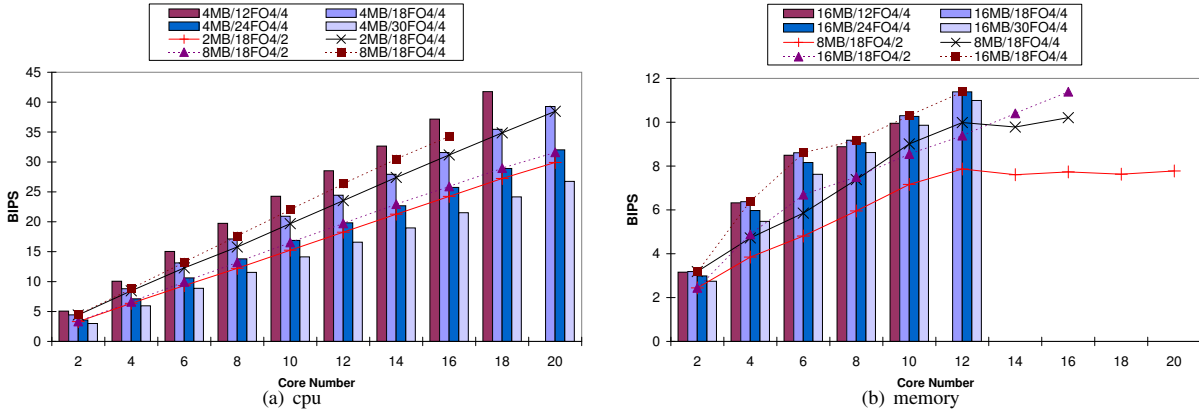


Figure 3. Performance of various configurations with chip area constraint at 400mm² (without thermal control) [1].

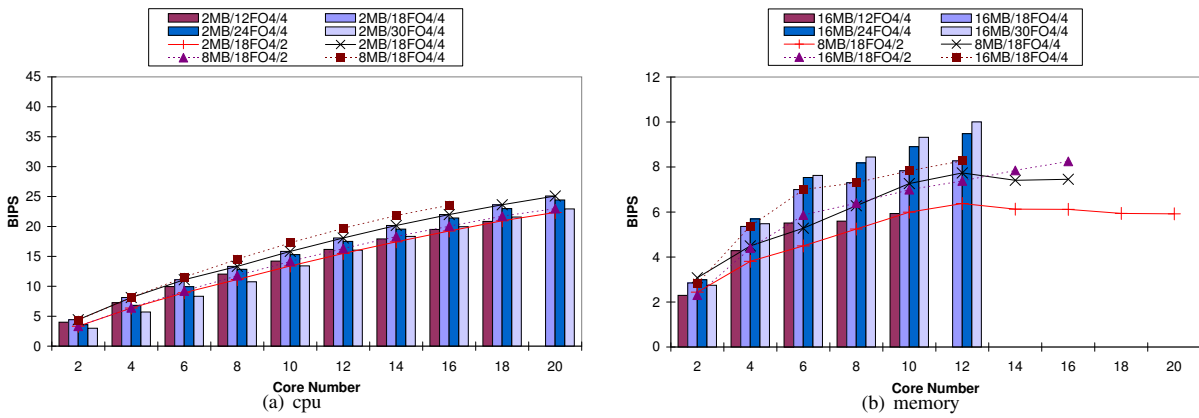


Figure 4. Performance of various configurations with chip area constraint at 400mm² (R = 0.1 heat sink). [1]

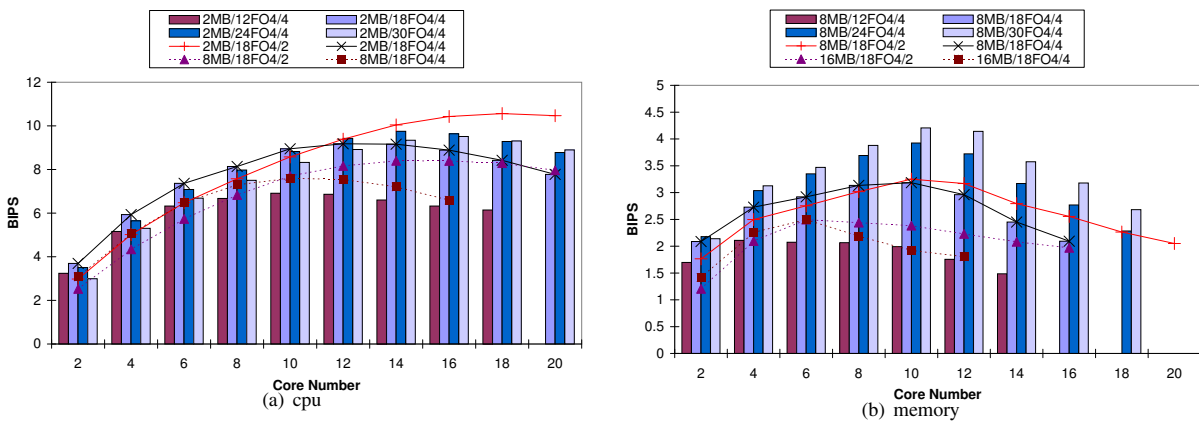


Figure 5. Performance of various configurations with chip area constraint at 400mm² (R = 0.5 heat sink). [1]

shallower pipelines that lower power dissipation, lower global temperature, and meet thermal constraints.

Simpler cores, characterized by shallower or narrower pipeline dimensions, tend to consume less power and, therefore, mitigate the core's thermal impact. In particular, the optimal pipeline depth shifts to 24 and 30FO4 delays per stage for CPU and memory-bound benchmarks, respectively, when comparing thermally-unconstrained to the most thermally constrained. Similarly, the optimal width shifts from an issue width of four (4W) to two (2W).

To better understand the impact of scaling the cores' depth vs. width, consider a baseline configuration 2MB L2, 18FO4, and 4W. As thermal constraints are imposed, the configuration may either shift to a shallower core (2MB/24FO4/4W) or shift to a narrower core (2MB/18FO4/2W). Since changes in width scale area for both functional units and many queue structures, whereas changes in depth only scale area for latches between stages, width reductions have a greater area impact relative to depth reductions. Thus, the 2MB/24FO4/4W core is a larger core relative to the 2MB/18FO4/2W and exhibits lower dynamic power density. However, the smaller 2MB/18FO4/2W core benefits from less leakage power per core and, consequently, less global power (since dynamic power dissipation is comparable for both cores).

From the simplified temperature models described earlier, total power output (P_{global}) has greater thermal impact for a chip with a poor heat sink (i.e., high thermal resistance, $R_{heatsink}$). Similarly, the thermal impact is dominated by the local power density, P_{core} , for a chip with a good heat sink. In this case, the transfer of heat from the silicon substrate to the spreader dominates thermal effects. Thus, to minimize chip heatup, it is advantageous to reduce width and global power in the context of a poor heat sink and advantageous to reduce depth and local power density in the context of a more expensive heat sink. This is most clearly seen with the CPU-bound benchmarks.

Hazards of Neglecting Thermal Constraints

Thermal constraints should be considered early in the design process. If a chip is designed without thermal constraints in mind, designers must later cut voltage and clock frequency to meet thermal constraints. The resulting voltage and frequency, and hence performance, will likely be cut more severely than if a thermally-aware configuration were selected from the beginning. The average difference between configurations designed from the beginning to account for thermal constraints versus those in which thermal constraints were only accommodated later ranges from 7 to 16%. There are notable exceptions, however. For example, for large, $400mm^2$ chips, omitting thermal consideration may result in huge performance degradations. In particular, the R=0.5 CPU- and memory-bound configurations result in a 40% – 90% difference in performance for BIPS. As area constraints are relaxed, the optimal point tends to include more cores and larger L2 caches. However, if the chip has severe thermal problems, voltage and frequency scaling must be aggressive to maintain thermal limits, moving into a region with significant non-linear

voltage and frequency scaling and producing large performance losses. For smaller chips with fewer cores and smaller L2 caches, the difference may be negligible because there are very few configurations to choose from. As future multi-core, server-class microprocessors target $400mm^2$ chips with more than eight cores, it will be essential to perform thermal analysis in the early-stages of the design process when decisions about the number and complexity of cores are being performed.

Voltage/Frequency Scaling versus Core Sizing

In meeting thermal constraints for large multi-core machines where global heat-up and total chip power is a concern, designers may be forced to choose among implementing fewer cores, smaller L2 caches, or employing aggressive voltage and frequency scaling (DVFS). Our results show DVFS superior to removing cores for CPU-bound applications as long as reductions in frequency are met by at least an equal reduction in dynamic and leakage power. Additional cores for CPU-bound applications provide linear increases in performance with near-linear increases in power dissipation. However, because of the strongly non-linear relationship between voltage scaling and clock frequency at low voltages, voltage scaling eventually stops providing super-linear power savings to make up for the performance (clock-frequency) loss. At this point, cores and L2 cache must be removed from the design to meet thermal constraints.

For example, a chip whose ratio of leakage to dynamic power is 3:7 no longer achieves super-linear power-performance benefit from DVFS scaling after reducing voltage by 55%; frequency of the chip drops to 18% and power dissipation also to 18% (dominated by leakage power, which only scales linearly with voltage). Further reductions in voltage lead to greater performance loss than power savings.

Figure 5 shows an example of this behavior with the 2MB/18FO4/4W design. When this design exceeds 14 cores, further increases in core count lead to performance degradation. Vdd scaling has exceeded 55%, and the additional DVFS scaling necessary to meet thermal constraints costs more performance than is gained by adding these additional cores. On the other hand, the 2MB/18FO4/2W design only requires Vdd scaling of 57% out to 20 cores, which is why this design is attractive even with the additional cores.

Similar analyses hold for memory-bound applications. In this case, the tradeoff is more complex, because the performance benefit from adding cores may be non-linear.

Accommodating Heterogeneous Workloads

Figures 3–5 also highlight the difficulty of accommodating a range of workload types under area constraints. This is less of a concern when looking at a small number of cores like most prior studies: for large numbers of cores, radically different configurations are possible.

CPU-bound and memory-bound workloads have different, incompatible optima. The performance loss from using the CPU-bound optimum with the memory-bound workload and vice-versa is severe, 37–41% and 26–53% respectively, depending on thermal constraints. Even if we try to identify com-

promise configurations, it is surprising how poorly they perform for one or the other workload. Of course, the best compromise depends on how heavily each workload is weighted. We tried to minimize the performance loss on both workloads.

As thermal limits become more severe, the difference among optimal configurations narrows, as the maximum possible number of cores and the L2 cache size is constrained, as the BIPS benefit of extra cores is reduced for CPU-bound benchmarks, and as the benefit of additional cache lines is reduced for memory-bound benchmarks.

CONCLUSIONS

This paper jointly optimized multi-core design to account for core count, core type, and operating voltage and frequency as a function of thermal constraints, and is based on work previously published in the computer-architecture field [1]. To accomplish this, a novel decoupled simulation approach was proposed that only uses detailed, cycle-by-cycle processor-pipeline simulation for individual cores, and then uses simpler and much faster cache-interaction simulation to model the multi-core performance. This in turn required a simple analytical model of localized, on-chip temperatures. Together, these results show the need for high-level thermal modeling techniques suitable for use in early-stage architecture studies, and how such thermal modeling capabilities can play a defining role in the organization of future, highly-integrated multi-core chips.

Joint optimization across multiple design variables is necessary. Even pipeline depth, typically fixed in architecture studies, may impact core area and power enough to change the optimal core count. Optimizing without thermal constraints and then scaling to a thermal envelope leads to dramatically inferior designs compared to those obtained from including thermal constraints in the initial optimization. For aggressive cooling solutions, reducing power density is at least as important as reducing total power. For low-cost cooling solutions, however, reducing total power is more important because raising power dissipation (even if power density is the same) raises a chip's temperature. In fact, thermal constraints appear to dominate other physical constraints like pin-bandwidth and power delivery. Once thermal constraints are met, at least within the design space studied here, power and throughput have been throttled sufficiently to fall safely within current off-chip I/O bandwidth capabilities and ITRS power-delivery projections.

Thermal constraints tend to favor shallower pipelines and narrower cores, and tend to reduce the optimal number of cores and L2 cache size. Nevertheless, even under severe thermal constraints, additional cores benefit throughput despite aggressive reductions in operating voltage and frequency. This is true until performance gains from an additional core is negated by the impact of the additional voltage and frequency scaling required of all the cores.

While the analytical approach described in this paper may suffice for homogeneous multi-core organizations, more sophisticated, yet extremely fast thermal modeling capabilities need to be developed in order to model heterogeneous multi-core organizations. An ability to capture transient, localized,

intra-core thermal phenomena without slowing down simulation time would certainly enhance multi-core design capabilities. Furthermore, an automated, parameterized way to explore different packaging choices in conjunction with exploration of multi-core organization and core-type is also needed. Finally, the ability to account for the impact of operating system scheduling and active-cooling choices is needed to round out a complete picture of temperature-aware multi-core design.

Acknowledgments

This work was funded in part by the National Science Foundation under grant nos. CAREER CCR-0133634, CAREER CCF-0448313, CCR-0306404, CCF-0429765, a Faculty Partnership Award from IBM T.J. Watson, a gift from Intel Corp., and an Excellence Award from the Univ. of Virginia Fund for Excellence in Science and Technology. A portion of Yingmin Li's work was performed during his internship at IBM T.J. Watson. The authors would also like to thank the anonymous reviewers for their helpful feedback.

References

- [1] Y. Li, B. Lee, D. Brooks, Z. Hu, and K. Skadron. CMP design space exploration subject to physical constraints. *Proc. of the Twelfth Int'l Symp. on High-Performance Computer Architecture*, Feb. 2006.
- [2] J. Huh, D. Burger, and S. W. Keckler. Exploring the design space of future CMPs. *Proc. of the Int'l Conf. on Parallel Architectures and Compilation Techniques*, Sep. 2001.
- [3] M. Ekman and P. Stenström. Performance and power impact of issue-width in chip-multiprocessor cores. *Proc. of the Int'l Conf. on Parallel Processing*, Oct. 2003.
- [4] P. Kongetira, K. Aingaran, and K. Olukotun. Niagara: A 32-way multi-threaded sparc processor. *IEEE Micro*, 25(2):21–29, Mar./Apr. 2005.
- [5] S. Chaudhry, P. Caprioli, S. Yip, and M. Tremblay. High performance throughput computing. *IEEE Micro*, 25(3):32–45, May/June 2005.
- [6] M. Moudgill, J. Wellman, and J. Moreno. Environment for powerpc microarchitecture exploration. *IEEE Micro*, 19(3), May/June. 1999.
- [7] D. Brooks, P. Bose, V. Srinivasan, M. Gschwind, P. Emma, and M. Rosenfield. Microarchitecture-level power-performance analysis: the powertimer approach. *IBM J. Research and Development*, 47(5), 2003.
- [8] D. Brooks, V. Tiwari, and M. Martonosi. Watch: a framework for architectural-level power analysis and optimizations. *Proc. of the 27th Int'l Symp. on Computer Architecture*, Jun. 2000.
- [9] B. Lee and D. Brooks. Effects of pipeline complexity on SMT/CMP power-performance efficiency. *Proc. of the Workshop on Complexity Effective Design*, Jun. 2005.
- [10] V. Zyuban. *Inherently lower-power high-performance superscalar architectures*. PhD thesis, Univ. of Notre Dame, Mar. 2000.
- [11] R. Kumar, V. Zyuban, and D. M. Tullsen. Interconnections in multi-core architectures: Understanding mechanisms, overheads and scaling. *The 32nd Int'l Symp. on Computer Architecture*, June. 2005.
- [12] K. Skadron, K. Sankaranarayanan, S. Velusamy, D. Tarjan, M. R. Stan, and W. Huang. Temperature-aware microarchitecture: Modeling and implementation. *ACM Trans. on Architecture and Code Optimization*, 1(1), Mar. 2004.
- [13] Y. Li, K. Skadron, Z. Hu, and D. Brooks. Performance, energy, and thermal considerations for SMT and CMP architectures. *Proc. of the Eleventh Int'l Symp. on High-Performance Computer Architecture*, Feb. 2005.
- [14] W. Huang, M. R. Stan, K. Skadron, S. Ghosh, K. Sankaranarayanan, and S. Velusamy. Compact thermal modeling for temperature-aware design. *Proc. of the 41st Design Automation Conf.*, June 2004.
- [15] S. Velusamy, W. Huang, J. Lach, M. R. Stan, and K. Skadron. Monitoring temperature in FPGA based SoCs. *Proc. of the 2005 Int'l Conf. on Computer Design*, 2005.