

# Throughput of Multi-core Processors Under Thermal Constraints\*

Ravishankar Rao, Sarma Vrudhula, and Chaitali Chakrabarti  
Consortium for Embedded Systems  
Arizona State University  
Tempe, AZ 85281, USA  
{ravirao, vrudhula, chaitali}@asu.edu

## ABSTRACT

We analyze the effect of thermal constraints on the performance and power of multi-core processors. We propose system-level power and thermal models, and derive expressions for (a) the maximum number of cores that can be activated, with and without throttling, (b) the speedup (multi-core over single core), and the total power consumption, both as functions of the number of active cores. These expressions involve parameters like power per core, thermal resistance of hottest die block and package, and leakage dependence on temperature. We also computed the above metrics (a) and (b) numerically by solving the detailed Hotspot circuit of an multi-core processor driven by a block-level exponential temperature-dependent leakage model. When compared to these numerical results, we found that the above expressions for (a) were at most 8% underpredicted, while those for (b) were accurately predicted. The proposed analytical approach is the first of its kind to relate metrics of interest in multi-core processors to high-level design parameters. Compared to numerical approaches, it provides much faster computation time, and valuable insight for processor designers.

## Categories and Subject Descriptors

B.8.2 [Performance and Reliability]: Performance Analysis and Design Aids; C.4 [Computer Systems Organization]: Performance of systems—*Modeling Techniques*

## General Terms

Design, Performance, Theory

## Keywords

multi-core processors, thermal management, speedup, throughput, power, leakage dependence on temperature

---

\*This work was supported by the Consortium for Embedded Systems at Arizona State University and by National Science Foundation grants ITR-0205227 and CNS-0509540.

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.

ISLPED'07, August 27–29, 2007, Portland, Oregon, USA.  
Copyright 2007 ACM 978-1-59593-709-4/07/0008 ...\$5.00.

## 1. INTRODUCTION

*Multi-core processors and challenges.* Technology scaling continues to make more transistors available within the same die area. Traditionally, these transistors made up a single processing unit. With each technology generation, the processor's performance improved significantly as a result of scaling the clock frequency. However, the cubic power-speed relationship and the resulting thermal issues made this scaling strategy unsustainable. This led processor designers to pursue alternate ways to utilize the extra transistors to get more performance for the same power. They found that by dividing the available transistors into multiple processing cores, which were each clocked lower than a comparable single-core processor, the total power consumption could be kept roughly the same while allowing increased total performance, especially for multi-threaded workloads [1]. Today, increasing the number of cores per die has become the new “scaling” strategy, with quad-core processors already on the market and massively multi-core architectures expected in the near future [2].

Multi-core processors however, present significant new challenges to processor designers, especially in terms of performance-power-thermal tradeoffs. As cooling and packaging technology does not scale with device technology, they continue to limit the total power consumption of a multi-core processor. This restriction on the power per core could require the maximum clock frequency per core to be reduced, which is detrimental to the performance of single-threaded workloads. Existing multi-core processors use a fixed maximum clock speed per core regardless of the number of active cores. This strategy is unable to utilize the power slack when all cores are not active. One solution that could be adopted in future multi-core processors is to make the maximum clock frequency a function of the number of active cores. For workloads with one or only a few threads, a small number of cores could be turned on and operated at large speeds. For more parallel workloads with many threads, more cores would be active, but they would each have to run at smaller speeds to meet the thermal constraints.

Amdahl's law suggests that the performance speedup (multi-core over single-core) of multi-threaded workloads does not scale linearly due to sequential code sections. But, the above discussion suggests that even for totally parallel workloads, the speedup may still not scale because of thermal constraints.<sup>1</sup> This then makes it very important to understand the relationship between metrics like speedup  $S$ , total power consumption  $P$  of a multi-core processor and the number of active cores  $N_a$ . It is also useful to quantify the

---

<sup>1</sup>We found that for sufficiently high power values, the speedup can actually decrease when many cores are activated.

effect of design parameters like package thermal resistance, floor-plan, leakage power per core, etc. on this relationship. Knowledge of this relationship would allow making early design choices (e.g. cooling technology, placement of hot blocks) that achieve a target performance and power profile. It would be particularly insightful if these relations  $S(N_a)$  and  $P(N_a)$  could be expressed analytically.

*Need for system-level models.* The nature of this relationship is however, not well understood even for the dual and quad-core processors on the market today. Designers mostly rely on projections obtained from detailed cycle-accurate power-performance-thermal simulation. This approach is very time consuming even for single-core processors, and is forbidding for multi-core processors. While it is indispensable for micro-architectural studies, it provides little analytical insight at the system level. Deriving closed form expressions like those mentioned above necessitates using simple power/performance/thermal models.

Researchers have proposed high-level models and used them to obtain analytical results or heuristics for thermal management [3–8]. However, most of these works did not address multi-core processors and they all use very simple power/thermal models which do not capture some or all of the following important effects: (a) the variations of power-density over different chip blocks, which leads to hotspots (see [9]), (b) leakage dependence on temperature (LDT), which is significant for 65 nm and beyond (see [5, 10]), and (c) the contrast in the thermal response times of the die and package. For example, [3–7] use a single thermal resistance and capacitance (which cannot model (c)), while [3, 4, 6, 7] modeled only the full-chip power (which ignores (a)), and [4, 6–8] neglect (b).

Architectural-level power models like [5, 11] and thermal models like Hotspot [9] are able to accurately model these effects, but they are better suited for simulation than analysis. For an  $N$ -core processor, Hotspot presents a thermal RC circuit of order  $20N+11$ . Further, the power consumption (input to the RC circuit system), is an exponential function of the die temperature (system state). To obtain the power and temperature of the die, one then needs to solve  $20N+11$  coupled non-linear differential equations, which cannot be done analytically even for  $N = 1$ . Hence, we need power/thermal models that account for effects (a)–(c) with reasonable accuracy, and scale easily for multi-core processors. Such models must also be technology-independent so they can be used, for example, to obtain performance projections for different processors in a technology roadmap. They must be simple enough to not need numerical methods. We propose models that meet these requirements.

*Static and dynamic thermal management.* Thermal management of processors has been studied mostly at the packaging and micro-architecture levels. With the former, the goal is to develop chip packages and cooling mechanisms that can safely dissipate a predefined power budget (called the thermal design power (TDP)) in the steady state. This design is inherently fixed; it is based on steady state analysis and does not adapt to run-time variations in processor power.<sup>2</sup> If some (combinations of) workloads consume more power than the TDP, the processor is dynamically throttled to keep the chip temperature below a safe threshold. This mechanism is called dynamic thermal management (DTM).

Micro-architectural studies like [9, 12, 13] have explored various DTM techniques like feedback control, local vs global voltage/frequency control, etc. They try to adapt the throttling to the time-varying processor workload to minimize the resulting perfor-

mance loss. Our work forms a middle ground between these approaches and could be considered static thermal management. We examine the effect of design parameters like the package thermal resistance, leakage power per core, etc. on metrics of interest like the (steady state) speedup and power consumption as a function of number of active cores. In computing these metrics, we account for both, the fixed relationship between power and temperature determined by the packaging, and the control of power consumption through processor throttling and turning off cores.

*Main contributions.* We studied existing detailed power [5] and thermal [9] models, made many careful approximations, and then extended them to a multi-core framework. The resulting system level power and thermal models account for effects (a)–(c) mentioned before, and yet, are simple enough to be analytically tractable, even for large numbers of cores. Using these models, we first developed a method to remove the circular dependency between leakage power and temperature. This allowed us to derive closed form expressions for the steady state temperature of each chip block and the package. We then derive the following expressions: (1) the maximum number of cores that can be activated, both with, and without throttling, in the steady state, (2) the steady state speedup and total power as a function of the number of active cores.

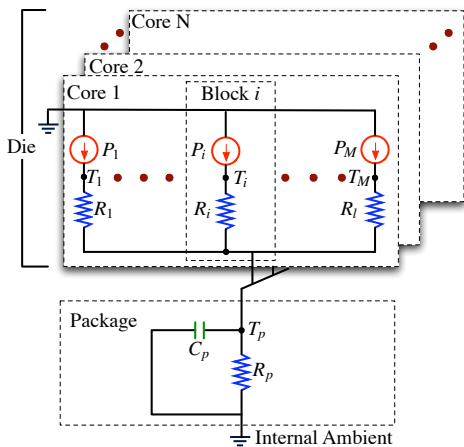
The above expressions contain several parameters of interest to a system designer like the static and dynamic power consumption (for hottest block and for full chip), the vertical thermal resistance of the hottest block, the leakage sensitivity to temperature, the chip threshold temperature and the ambient temperature. We setup a simulation framework to compute the steady-state speedup and power of a multi-core processor using the low-level power and thermal models from [5, 9]. We then compared the resulting metrics to that predicted by the proposed closed form expressions, and found good agreement (worst case error: 8%). The proposed model accurately accounts for the important power/thermal effects, but has the advantage of analytical insight and faster computation speed.

## 2. SYSTEM MODEL

We model a multi-core processor as a set of  $N$  identical cores on a die, with each core having  $M$  functional unit blocks. Like Hotspot [9] we assume that each block has uniform power density over its die surface. Multi-core processors are often marketed as offering up to  $N$  times greater throughput for jobs that can be parallelized. However, we show that thermal limits can negatively affect speedup, and cause the maximum speedup to be less than  $N$ , even for completely parallel workloads. To demonstrate this behavior, we assume our workload consists of identical independent threads, with each active core of the processor executing one thread. For such a workload, we define the speedup  $S(N_a)$  as the ratio of the throughput (total retired instructions per second over all cores) achieved with  $N_a$  active cores, to that with one active core.

As our interest is in steady state performance and power, average values are assumed for the instructions per clock cycle (IPC) and power consumption of each job. Since the cores are identical and execute identical threads, their steady state power and thermal profiles would be approximately the same. Hence, we assume that each active core is throttled by the same amount (global throttling policy) to satisfy thermal constraints. We chose dynamic frequency scaling (DFS) as the throttling mechanism, for which, the steady state throughput and dynamic power per core are proportional to the normalized clock speed  $s$ . As this is also true for most other throttling mechanisms [13, 14] like clock duty ratio variation and

<sup>2</sup>Even variable speed cooling fans will have to operate at full speed if power consumption reaches TDP.



**Figure 1: High-level thermal model of a multi-core processor.**

fetch throttling<sup>3</sup>, the assumption of DFS is quite general. The normalized clock speed  $s$  is assumed to be continuously variable from a specified minimum speed  $s_{\min}$  to 1. As the amount of throttling varies with the number of active cores,  $s$  is a function of  $N_a$ . Finally, assuming that no throttling will be needed for  $N_a = 1$ , the speedup  $S$  when  $N_a$  cores are active is given by  $N_a s(N_a)$ .

**Thermal model.** We developed a high-level thermal model based on the Hotspot thermal RC circuit model [15]. Hotspot uses an analogy between electrical circuit phenomena and heat transfer phenomena. It divides the processor die and package into multiple blocks. The flow of heat between the blocks modeled through connecting thermal resistances, and the heat storage in the blocks by thermal capacitances. The power consumed by each chip block (which typically corresponds to a functional unit) is modeled by a current source. The chip bottom connects to the package (modeled with 10 blocks), which then connects to the convective cooling mechanism (a single block). For a processor with 20 functional blocks, the resulting thermal equivalent circuit is of order  $20N + 11$  (equals 171 for eight cores).

We analyzed the parameters of the Hotspot equivalent circuit for the Alpha 21264 floorplan, and we found that: (1) The thermal resistance  $R_{\text{conv}}$  and capacitance  $C_{\text{conv}}$  of the cooling system are much larger than that of the other units in the package. Hence, we modeled the package and cooling system as a single lumped block. (2) The lateral resistances between chip units were at least four times larger than the vertical resistances that connected chip units to the package, i.e. most of the heat flows vertically in the silicon die. Hence, we neglect lateral resistances on chip. Note that this assumption does not eliminate the occurrence of hotspots/thermal gradients as they are mainly caused by differences in power density among chip blocks, and are affected to a lesser extent by lateral heat flow between them. (3) As our work focusses on steady state conditions, we ignore all thermal capacitances. However, we did use them to remove the circular dependency between leakage power and die temperature, which we discuss later.

Figure 1 shows the resulting reduced-order thermal RC circuit. The package and cooling system (henceforth called the package) is modeled by a first order RC circuit, with the resistance and capacitance values  $R_p$  and  $C_p$  obtained by a model order reduction of the corresponding Hotspot blocks. Each chip block  $i$  is modeled

<sup>3</sup>The exception is Dynamic voltage and frequency scaling (DVFS), which is often not preferred for DTM [14].

by a current source  $P_i$  (representing its power consumption) and a thermal resistance  $R_i$  (representing conduction vertically). The values for  $R_i$  were set to the vertical resistance connecting each chip block to the spreader top in the original Hotspot circuit. The voltage across the capacitance  $C_p$  ( $C_i$ ) is the temperature (relative to the internal ambient) of the lumped package  $T_p$  ( $i^{\text{th}}$  chip block  $T_i$ ).

**Power model.** The power consumed  $P_i$  by block  $i$  of an active core has two parts, dynamic and static. The dynamic power at speed  $s$  is given by  $sP_{d,i}$ , where  $P_{d,i}$  is the dynamic power at full speed  $s = 1$ . The static power  $P_{s,i}$  is, in general, an exponentially increasing function of the temperature of the block  $T_i$  [5, 11]. Thus, the block power is a function  $P_i(s, T_i)$  of the core speed and block temperature. To simplify our analysis, we approximate the nonlinear relationship  $P_{s,i}(T_i)$  with a 2-piece linear function:

$$P_{s,i}(T_i) = \begin{cases} P_{s,i,\text{mid}} - k_{i,1}(T_{\text{mid}} - T_i), & 0 \leq T_i \leq T_{\text{mid}}, \\ P_{s,i,\text{max}} - k_{i,2}(T_{\text{max}} - T_i), & T_{\text{mid}} < T_i \leq T_{\text{max}}. \end{cases} \quad (1)$$

Here, the 0 temperature limit represents the internal ambient, and the  $T_{\text{max}}$  limit, the maximum permissible die temperature.  $T_{\text{mid}}$  is a temperature lying between these extremes.  $P_{s,i,\text{mid}}$  and  $P_{s,i,\text{max}}$  are the static power values  $P_{s,i}(T_i)$  at  $T_{\text{mid}}$  and  $T_{\text{max}}$ , respectively. Figure 2 shows the total power consumption as a function of die temperature of a single core processor for 65 nm technology computed using the power models in [5]. For one of the profiles, the corresponding linear approximation is also shown for comparison.

To determine which of the two regions to use in computing  $P_{s,i}$ , one must know the block temperature  $T_i$ , but this again depends on  $P_{s,i}$ . We show in Section 3 how this circular dependency can be removed to compute both  $P_{s,i}$  and  $T_i$  in terms of the coefficients  $k_i$ . The resulting expressions are used to compute  $P_{s,i}$  and  $T_i$  by first assuming that  $T_i$  is in a particular region (say  $(T_{\text{mid}}, T_{\text{max}}]$ ) and using the corresponding coefficients ( $k_{i,2}, P_{s,i,\text{max}}$ ). If the calculated value of  $T_i$  is in fact in the other region,  $T_i$  and  $P_{s,i}$  are recomputed using the coefficients of that region ( $k_{i,1}, P_{s,i,\text{mid}}$ ). In the interest of clarity and brevity, the rest of the analysis assumes that all die temperatures are in the range  $(T_{\text{mid}}, T_{\text{max}}]$ . It is to be understood that the procedure mentioned above is to be used to determine the correct set of coefficients in each of the resulting expressions.

### 3. PRELIMINARY RESULTS

In this section, we obtain preliminary results, which we will then use to find the main results in Section 4. First, we remove the circular dependency between leakage power and temperature by expressing both of them in terms of the package temperature. We then redefine certain groups of parameters into ‘‘apparent’’ static and dynamic powers and use them to compute the steady state package temperature. Finally, we show how the hottest block in a chip can be computed statically for a given power profile.

**Decoupling leakage and chip temperature.** From the thermal RC circuit in Figure 1,

$$C_i \frac{dT_i(t)}{dt} = -\frac{T_i(t) - T_p(t)}{R_i} + [sP_{d,i} + P_{s,i,\text{max}} - k_i(T_{\text{max}} - T_i)].$$

For the Alpha 21264 thermal circuit, the largest time constant for any chip block was 10 ms while that of the package was 57 s. This is true in general for most processors, because the small die volume is unable to store as much heat as the large package, and hence responds faster to changes in power. Due to this relatively large package time constant,  $T_p(t)$  changes much slower than  $T_i(t)$ . So, we can assume  $T_p$  to be a constant parameter and solve the above

first order linear differential equation in closed form to get  $T_i(t) = T_i(0)e^{-\beta_i t} + (\alpha_i/\beta_i)(1 - e^{-\beta_i t})$ , where  $\alpha_i = (1/C_i)(P_{s,i,\max} - k_i T_{\max} + sP_{d,i} + T_p/R_i)$  and  $\beta_i = (1/C_i)(1/R_i - k_i)$ . If  $k_i > 1/R_i$  for any chip block, that block will experience *thermal runaway*. Assuming the chip design ensures no thermal runaway for  $T_i \leq T_{\max}$ , we proceed to rewrite the expression for  $T_i$  as follows

$$T_i = \alpha_i/\beta_i = \zeta_i T_p + (P'_{s,i} + sP'_{d,i})R_i, \text{ where} \quad (2)$$

$\zeta_i \triangleq 1/(1 - k_i R_i)$  (leakage power factor),  $P'_{s,i} \triangleq \zeta_i (P_{s,i,\max} - k_i T_{\max})$  (apparent static power) and  $P'_{d,i} \triangleq \zeta_i P_{d,i}$  (apparent dynamic power).

Substituting for  $T_i$  in the earlier expression for  $P_{s,i}$ , we now get the total power for block  $i$  independent of its temperature  $T_i$ :

$$P_i = P'_{s,i} + sP'_{d,i} + (\zeta_i - 1)T_p/R_i. \quad (3)$$

The die static power still depends on temperature, but of the common package instead of the respective block's temperature.

**Steady state package temperature.** We are now ready to compute the package temperature  $T_{p,ss}$  in steady state assuming  $N_a$  cores are operating at speed  $s$ . We first define the total apparent static and dynamic power per core as  $P'_s \triangleq \sum_{i=1}^M P'_{s,i}$ ,  $P'_d \triangleq \sum_{i=1}^M P'_{d,i}$ . Also, we define  $G \triangleq \sum_{i=1}^M (\zeta_i - 1)/R_i$  which has units of thermal conductance. We then have from Figure 1 that  $T_{p,ss} = (N_a \sum_{i=1}^M P_i) R_p = N_a R_p (P'_s + sP'_d + GT_{p,ss})$ . Rearranging, we have

$$T_{p,ss} = \frac{N_a}{1 - GR_p N_a} (P'_s + sP'_d) R_p. \quad (4)$$

**Identifying the hottest chip block.** Using (2) and (4), we obtain the steady state die temperature for block  $i$  as

$$T_{i,ss} = (P'_{s,i} + sP'_{d,i})R_i + \frac{N_a \zeta_i}{1 - GR_p N_a} (P'_s + sP'_d) R_p, \quad (5)$$

which is a function of the global core speed  $s$ . The hottest block  $h$  is defined as the one that maximizes  $T_{i,ss}$  over  $i = 1, \dots, M$ . In general, different blocks within a given core can become the hottest depending on the speed. However, we have found that the block that is hottest at  $s = 1$  remains the hottest at all speeds, for the following reason. Typically, if block  $i$  has greater static power than block  $j$ , it also has greater dynamic power (at  $s = 1$ ). This tends to maintain the same ordering among chip blocks in terms of total power (and hence temperature) even if the speed  $s$  is varied.

## 4. MAIN RESULTS

In this section, we first obtain the maximum number of cores that can be activated, both with throttling  $N_m$  and without throttling  $N_c$ . For the case  $N_a > N_c$  where throttling is required, we then compute the required speed  $s^*$  required to run  $N_a$  cores simultaneously within the thermal limit. This gives us the total speedup  $S = N_a s^*$ . Finally, we derive the total power  $P(N_a)$ .

**Maximum active cores without throttling.** When  $N_c$  cores are running at full speed  $s = 1$ , we must still have the hottest chip block temperature lower than  $T_{\max}$ . Using (5), we have

$$T_{h,ss} = \left( P'_{s,h} + P'_{d,h} \right) R_h + \frac{N_c \zeta_h}{1 - GR_p N_c} (P'_s + P'_d) R_p \leq T_{\max}.$$

We then solve the above inequality for  $N_c$ :

$$N_c = \left\lfloor \frac{T_{\max} - \left( P'_{s,h} + P'_{d,h} \right) R_h}{\zeta_h R_p (P'_s + P'_d) + \left[ T_{\max} - \left( P'_{s,h} + P'_{d,h} \right) R_h \right] GR_p} \right\rfloor. \quad (6)$$

In the above equation,  $\lfloor \cdot \rfloor$  is the floor function. We will henceforth call  $N_c$  the *critical number of active cores*.

**Maximum active cores with throttling.** In a well-designed multi-core processor, all  $N$  cores can be activated if they are sufficiently throttled. For certain power/thermal parameters (e.g. high leakage), this may actually not be possible. We now obtain the maximum number of cores  $N_m$  that can be activated with throttling. When  $N_m$  cores are running at the least supported speed  $s_{\min}$ , we must have the hottest chip block temperature lower than  $T_{\max}$ :

$$T_{h,ss} = \left( P'_{s,h} + s_{\min} P'_{d,h} \right) R_h + \frac{N_m \zeta_h}{1 - GR_p N_m} (P'_s + s_{\min} P'_d) R_p \leq T_{\max}.$$

We then solve the above inequality for  $N_m$ :

$$N_m = \left\lfloor \frac{T_{\max} - \left( P'_{s,h} + s_{\min} P'_{d,h} \right) R_h}{\zeta_h R_p (P'_s + s_{\min} P'_d) R_p + \left[ T_{\max} - \left( P'_{s,h} + s_{\min} P'_{d,h} \right) R_h \right] GR_p} \right\rfloor.$$

Processor designers can use estimated power/thermal parameters to compute  $N_m$  and ensure that it is no smaller than  $N$ .

**Speedup with  $N_a$  active cores.** For all  $N_a > N_c$ , some throttling will be required, and to maximize the throughput, the throttled speed  $s^*$  should be chosen just small enough to keep the hottest block  $h$  of each core below  $T_{\max}$ . We then have  $T_{h,ss} = T_{\max}$ , when  $s = f$  (say), or, using (5):

$$T_{h,ss} = T_{\max} = \left( P'_{s,h} + f P'_{d,h} \right) R_h + \frac{N_a \zeta_h}{1 - GR_p N_a} (P'_s + f P'_d) R_p.$$

We can then solve the above equation for  $f$  to get

$$f(N_a) \triangleq \frac{T_{\max} - P'_{s,h} R_h - N_a \left[ \zeta_h R_p P'_s + \left( T_{\max} - P'_{s,h} R_h \right) GR_p \right]}{P'_{d,h} R_h + N_a \left[ \zeta_h R_p P'_d - P'_{d,h} R_h GR_p \right]}. \quad (7)$$

Combining this result with the no throttling case, we have

$$s^*(N_a) = \begin{cases} 1, & N_a \leq N_c, \\ f(N_a), & N_a > N_c. \end{cases} \quad (8)$$

The speedup is then simply  $S(N_a) = N_a s^*(N_a)$ .

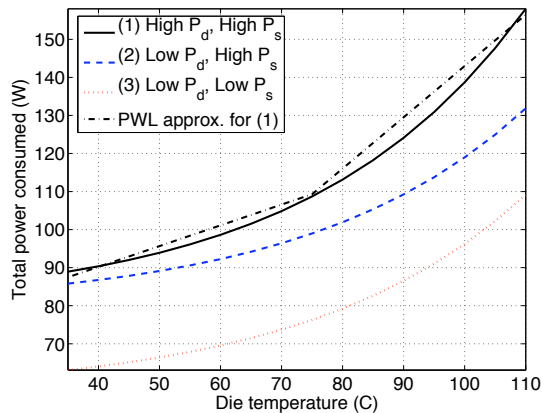
**Total power with  $N_a$  active cores.** Suppose  $N_a$  cores are operating at speeds  $s^*(N_a)$ . We use (3) and (4) to find the power consumed by block  $i$  in each core:

$$P_i(N_a) = P'_{s,i} + s^*(N_a) P'_{d,i} + \frac{\zeta_i - 1}{R_i} \left[ \frac{N_a (P'_s + s^*(N_a) P'_d) R_p}{1 - GR_p N_a} \right]$$

The total power consumed over all active cores and blocks is  $P(N_a) = N_a \sum_{i=1}^M P_i(N_a)$ . After some algebra,  $P(N_a)$  reduces to

$$P(N_a) = N_a \frac{P'_s + s^*(N_a) P'_d}{1 - GR_p N_a} = \left[ \frac{T_{p,ss}(s)}{R_p} \right]_{s=s^*(N_a)}. \quad (9)$$

We note from (9) that for  $N_a \leq N_c$ ,  $s^*(N_a) = 1$ , and  $P(N_a)$  is proportional to  $N_a/(1 - GR_p N_a)$ , which can be shown to be an increasing



**Figure 2: Power profiles  $P_1(T)$  of a single-core processor for different values of static and dynamic power.**

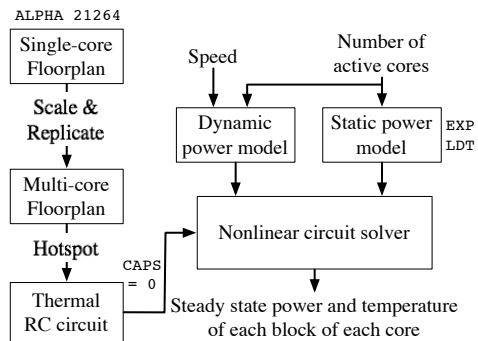
convex function of  $N_a$  (for  $N_a < 1/(GR_p)$ ). This is unlike the total throughput  $S(N_a)$  which is linear in  $N_a$  for  $N_a \leq N_c$ . This is because increasing the number of active cores results in a greater package temperature due to (4), which then causes each core’s leakage power to increase due to LDT according to (3).

## 5. RESULTS

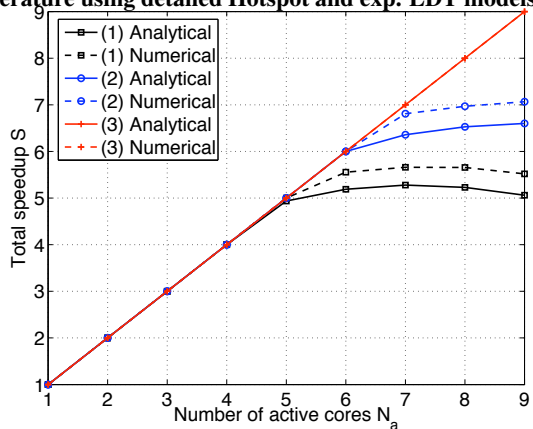
*Model data.* We took the single-core Alpha 21264 EV6 floorplan from [9], scaled its dimensions to create  $N$  copies that fit within the same area as the original single-core processor. For this  $N$ -core floorplan, Hotspot [9] computes the equivalent thermal RC network. We then applied the approximations mentioned in Section 2 to this Hotspot circuit and obtained our high-level thermal model. We set the convection thermal resistance in Hotspot to  $0.3 \text{ W}/^\circ\text{C}$  to reflect the power-thermal characteristics of modern server-grade processors [16, 17]. This lower value represents improved packaging and cooling technology and was chosen to allow a chip maximum temperatures  $T_{\max} = 110^\circ\text{C}$ , and a steady-state power consumption (with throttling) of about 130 W.

Given a floorplan of a future multi-core processor, the formulas for computing thermal resistances are well-known because they depend on properties (like thermal conductivity) that do not change much with technology scaling. However, obtaining block-level power data (especially leakage) for future multi-core processors is much harder because it depends on many technology-dependent parameters (like subthreshold leakage per transistor). Hence, we use the following general approach. First, we obtain leakage and dynamic power data for different functional blocks using the temperature-aware models from [5]. We then scale the leakage power (function of temperature) and dynamic power (function of speed), keeping relative block numbers the same, to obtain power numbers  $P_1(T, s)$  representative of a modern single-core server processor.

We then model the *total* power consumption of a “corresponding”  $N$ -core processor with all cores active to be  $P_N(T, s) = k_p P_1(T, s)$ . The scaling factor  $k_p$  is a simple and general way to model the changes in chip power consumption over different technologies. We then compute the per-core power consumption as  $P_N(T, s)/N$ . Figure 2 shows a plot of the total power consumption of a single-core processor  $P_1(T, 1)$ , i.e. with all cores running at full speed. To study the effect of dynamic and static powers on multi-core speedup and power, we construct three profiles: (1) high dynamic and static power, (2) low dynamic, but high static power, and (3)



**Figure 3: Numerical computation of steady state power and temperature using detailed Hotspot and exp. LDT models.**



**Figure 4: Total throughput vs number of active cores.**

low dynamic and static power. We note here that we chose  $k_p = 1.5$ ,  $T_{\text{mid}} = 75^\circ\text{C}$  (see Figure 2),  $s_{\min} = 0.1$  for our setup.

*Total throughput and power.* We now compute the throughput and power curves for a 9-core processor represented by the above power-thermal data. We do this first using the closed form expressions for  $S$  and  $P$  obtained in Section 5, and then verify these results numerically by using the detailed Hotspot and exponential LDT power models. Figure 3 describes the method used to obtain the steady state power and temperature of each block of each core numerically, for a given throttling speed  $s$  and number of active cores  $N_a$ .

The Hotspot thermal RC circuit of the multi-core processor is first pruned of all capacitances (as we are interested in steady state solution). This resistance network, when combined with the exponential LDT static power model and the (constant) dynamic power model constitutes a non-linear system because the current sources in each die block depend exponentially on the voltage of that block. To obtain the steady state values of these currents (power consumption) and voltages (temperature), we use a non-linear equation solver that solves the 191 KCL equations for each node in the circuit. To compute the throttling speed  $s^*(N_a)$  numerically, we repeat the procedure described above for different speeds over the range  $[s_{\min}, 1]$  by performing a binary search, and select the speed that keeps the hottest block on the die just below  $T_{\max}$  in steady state. Once we find  $s^*(N_a)$ , the output of the solver gives us the power consumption of each block, which is summed to get  $P(N_a)$ .

Figure 4 shows the speedup of a 9-core processor as a function of the number of active cores, computed both analytically and nu-

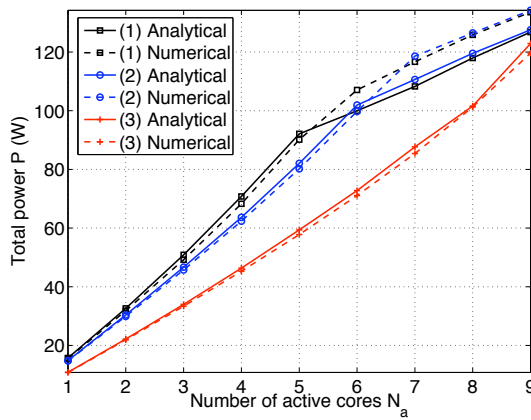


Figure 5: Total power vs number of active cores.

merically for each of the profiles in Figure 2. For the profile (1), the speedup scales linearly until  $N_c = 4$  cores, beginning to slow down slightly for  $N_a = 5$ , reaching a maximum at  $N_a = 7$  and decreasing for  $N_a = 8$  and 9. As the static power is high, and because throttling the processor only reduces the dynamic power linearly, a lot of throttling is needed when the number of active cores (and hence total static power) grows. This speedup curve is unacceptable as using all cores results in less performance than using only seven cores, which defeats the purpose of deploying nine cores.

If the dynamic power were to be reduced, as in profile (2), we see that the speedup now scales linearly up to  $N_c = 6$  cores. Although it veers off the straight line for  $N_a > 6$ , it continues to increase. This allows utilizing all  $N$  cores, with increasing speedup for increasing  $N_a$ . The improvement over (1) is because the reduced dynamic power requires lower throttling to meet the power-thermal budget. If the static power were also reduced sufficiently, as in profile (3), the total power is within the thermal limit, the speedup curve becomes the ideal straight line, and all cores can be simultaneously run at full speed. We see that the analytical results for  $N_c$  match the numerical results exactly for all three profiles. The speedup numbers predicted analytically are slightly underpredicted for  $N_a > N_c$ , with the worst case error being 8%. For profile (1), the analytical results took 71 ms to compute versus 96 s for the numerical.

We believe that using the proposed approach provides a good tradeoff between analytical insight and numerical accuracy. For example, suppose a designer wanted to deploy better packaging or cooling to have the speedup curve for profile (2) look like that for profile (3). With the numerical approach, the designer would have to try different values of various package and cooling parameters in Hotspot, and redo the procedure described in Figure 3. With the analytical approach, the designer could simply set  $N_c = 9$  in (6) and solve for  $R_p$ . The expressions in Section 5 also allow one to easily compute the sensitivity of variables of interest to design parameters (like  $T_{max}$ ) by simply taking the partial derivative, e.g. the speedup is a linear function of  $T_{max}$ , while  $N_c$  is inversely proportional to the total (apparent) power consumption per core.

Figure 5 shows the total power consumed by the same 9-core processor as the number of active cores is varied. As predicted in Section 5 all the power profiles have corresponding  $P(N_a)$  curves that are strictly convex and increasing for  $N_a \leq N_c$ . Also, for  $N_a > N_c$ , they are concave and increasing. The prediction error w.r.t. the numerical results are small for  $N_a \leq N_c$ ; the error here is mainly due to the piecewise-linear approximation for LDT as both approaches predict the throttling speed correctly as 1. For  $N_a > N_c$ , the error

in computing the throttling speed causes a corresponding underprediction (worst case 7%) in computing the dynamic power.

## 6. CONCLUSION

The processor industry has plans to aggressively scale the number of cores on a die. There is a great need for tools and models at all design levels to understand the relationships between performance, power and heat for such processors. We propose high-level thermal and power models that incorporate important effects like leakage dependence on temperature (LDT), and die power density variations. Using a linear approximation for LDT, we removed the circular dependency between the power and temperature of a chip block. We developed simple closed form expressions for (a) the speedup and power of an  $N$ -core processor as a function of the number of active cores, (b) the maximum number of cores that can be activated, both, with and without throttling. These expressions contain parameters related to macro-level system power and thermal parameters which can be used for early design exploration. We validated these results by comparing them to the output of a non-linear circuit solver, which computes the steady state of the Hotspot thermal circuit driven by an exponential LDT power model.

## 7. REFERENCES

- [1] D. Geer, "Chip makers turn to multicore processors," *IEEE Computer*, vol. 38, no. 5, pp. 11–13, May 2005.
- [2] J. Held, J. Bautista, and S. Koehl. (2006) From a few cores to many: A tera-scale computing research overview.
- [3] S. Heo, K. Barr, and K. Asanovic, "Reducing power density through activity migration," in *Proc. Intl' Symp. Low Power Electronics and Design (ISLPED)*, 2003, pp. 217–222.
- [4] A. Cohen, F. Finkelstein, A. Mendelson, R. Ronen, and D. Rudoy, "On estimating optimal performance of cpu dynamic thermal management," *IEEE Computer Architecture Letters*, vol. 2, no. 1, pp. 6–6, 2003.
- [5] W. Liao, L. He, and K. M. Lepak, "Temperature and supply voltage aware performance and power modeling at microarchitecture level," *IEEE Trans. Computer-Aided Design*, vol. 24, no. 7, pp. 1042–1053, July 2005.
- [6] W. Lee, K. Patel, and M. Pedram, "Dynamic thermal management for mpeg-2 decoding," in *Proc. Intl' Symp. Low Power Electronics and Design (ISLPED)*, 2006, pp. 316–321.
- [7] R. Rao, S. Vrudhula, C. Chakrabarti, and N. Chang, "An optimal analytical solution for processor speed control with thermal constraints," in *Proc. Intl' Symp. Low Power Electronics and Design (ISLPED)*, October 2006, pp. 292–297.
- [8] R. Mukherjee and S. O. Memik, "Physical aware frequency selection for dynamic thermal management in multi-core systems," in *Proc. Intl' Conf. Computer Aided Design (ICCAD)*, 2006, pp. 547–552.
- [9] W. Huang, M. R. Stan, K. Skadron, K. Sankaranarayanan, and S. Ghosh, "Hotspot: A compact thermal modeling method for cmos vlsi systems," *IEEE Trans. VLSI Sys.*, vol. 14, no. 5, pp. 501–513, May 2006.
- [10] H. Su, F. Liu, A. Devgan, E. Acar, and S. Nassif, "Full chip leakage estimation considering power supply and temperature variations," in *Proc. Intl' Symp. Low Power Electronics and Design (ISLPED)*, 2003, pp. 78–83.
- [11] Y. Li, D. Parikh, Y. Zhang, K. Sankaranarayanan, M. Stan, and K. Skadron, "State-preserving vs. non-state-preserving leakage control in caches," in *Proc. Design Automation and Test in Europe Conf. (DATE)*, 2004, pp. 22–27.
- [12] C. Isci, A. Buyuktosunoglu, C.-Y. Cher, P. Bose, and M. Martonosi, "An analysis of efficient multi-core global power management policies: Maximizing performance for a given power budget," in *Proc. Intl' Symp. Microarch. (MICRO)*, 2006, pp. 347–358.
- [13] J. Donald and M. Martonosi, "Techniques for multicore thermal management: Classification and new exploration," in *Proc. Intl' Symp. Comp. Arch. (ISCA)*, 2006.
- [14] D. Brooks and M. Martonosi, "Dynamic thermal management for high-performance microprocessors," in *Intl' Symp. High Speed Computer Arch. (HPCA)*, 2001, pp. 171–182.
- [15] K. Skadron, T. Abdelzaher, and M. R. Stan, "Control-theoretic techniques and thermal-RC modeling for accurate and localized dynamic thermal management," in *Proc. Intl' Symp. High Perf. Comp. Arch. (HPCA)*, 2002, pp. 17–28.
- [16] AMD Athlon 64 Processor Power and Thermal Data Sheet, Advanced Micro Devices, August 2004.
- [17] Intel Pentium 4 Processor 6x1 Sequence: Datasheet, Intel Corp., January 2006.