Physical Aware Frequency Selection for Dynamic Thermal Management in Multi-Core Systems

Rajarshi Mukherjee[¶] and Seda Ogrenci Memik
[¶]Synopsys, Inc. Mountain View, CA 94043
Department of Electrical Engineering and Computer Science
Northwestern University, Evanston, IL 60208

ABSTRACT

In order to maintain performance per Watt in microprocessors, there is a shift towards the chip level multiprocessing paradigm. Microprocessor manufacturers are experimenting with tens of cores, forecasting the arrival of hundreds of cores per single processor die in the near future. With such large-scale integration and increasing power densities, thermal management continues to be a significant design effort to maintain performance and reliability in modern process technologies. In this paper, we present two mechanisms to perform frequency scaling as part of Dynamic Frequency and Voltage Scaling (DVFS) to assist Dynamic Thermal Management (DTM). Our frequency selection algorithms incorporate the physical interaction of the cores on a large-scale system onto the emergency intervention mechanisms for temperature reduction of the hotspot, while aiming to minimize the performance impact of frequency scaling on the core that is in thermal emergency. Our results show that our algorithm consistently succeeds in maximizing the operating frequency of the most critical core while successfully relieving the thermal emergency of the core. A comparison of our two alternative techniques reveals that our physical aware criticality-based algorithm results in 11.7% faster clock frequencies compared to our aggressive scaling algorithm. We also show that our technique is extremely fast and is suited for real time thermal management

Categories and Subject Descriptors: J.6 [**Computer Applications**]: Computer-aided Engineering (CAD):

General Terms: Algorithms, Performance, Experimentation.

Keywords: Dynamic Thermal Management, Multi-Core System.

1. INTRODUCTION

Microprocessor designs will continue to rely on technology scaling to meet aggressive performance/area targets. While advances in process technology will enable architectural innovations such as multi-threading, multi-core processors, aggressive execution techniques, and advanced memory management, they are expected to exacerbate existing hurdles in processor design and introduce a series of new challenges. Foremost, steady miniaturization and large-scale integration leads to increasing power densities [1, 2]. Power dissipated on a chip is converted to heat, which causes localized heating resulting in the formation of hotspots [3]. This can create reliability threats and also impact performance.

An active area in thermal-aware high performance microprocessor system design is Dynamic Thermal Management (DTM). Intel Pentium 4, Pentium M, and IBM PowerPC processors are equipped with temperature sensors that trigger alerts if temperature exceeds above a specified limit and processor activity and power consumption is regulated [4-6]. DTM mainly involves detecting a pre-defined thermal emergency level T_O at which the processor is throttled. Once the temperature is below a pre-defined reset temperature T_R the throttling mechanism is disabled. There is a target cooling period τ within which the core temperature should be reduced from T_O to T_R.

In this paper, we present a frequency selection algorithm to assist DTM for multi-core systems. Our goal is to develop a systematic approach that can help determine the best operating frequency level for the core in thermal emergency during performance throttling. This entails to bring this core out of thermal emergency within a given time period while maintaining its operating frequency as high as possible to impact its performance minimally. Our main contribution is our novel approach to the frequency selection, which presents two important capabilities. First, it considers physical characteristics of the system, i.e. the thermal interaction between physically adjacent cores during thermal emergency management. Second, it offers a fast technique for an optimization scheme to minimize the negative impact of frequency throttling onto the performance of the core that is in thermal emergency.

Various techniques exist for run-time thermal management such as clock frequency scaling, dynamic voltage and frequency scaling (DVFS), clock gating, and migrating computation. Powell et al. [7] proposed heat and run thread migration in chip multiprocessor (CMP). Huang et al. [8] proposed a framework for dynamic energy efficiency and temperature management. Evaluation of thermal efficiency of SMT and CMP architectures have also been studied [9, 10]. Various techniques exist for run-time thermal management [11]. In previous studies [8, 11] results have been shown for a fixed frequency throttling level determined a priori. Our work differs in the fact that we aim to identify optimal frequency levels to be used by the DVFS scheme.

Voltage and frequency scaling for power reduction has been implemented in Crusoe processor [12]. Both online and offline DVFS schemes have targeted to scale frequency to match performance demand and optimize energy. They do not address thermal management. Relieving thermal stress of the system is critical and takes precedence over performance in thermal emergencies, however this trade-off can be made in a systematic manner to minimize the performance impact, which is our goal.

We propose a mechanism to perform frequency scaling (with associated voltage scaling necessary to maintain the required operating frequency) for the core exhibiting emergency temperature T_O to assist its dynamic thermal management. We refer to such a core as *hot core* or *hotspot*. The frequency is selected such that the temperature of the hot core is reduced to reset temperature T_R in the target cooling period τ. To the best of

[¶]This work was done while the author was at Northwestern University.

ICCAD'06, November 5-9, 2006, San Jose, CA

our knowledge no technique has been proposed to perform physical aware frequency selection for thermal management. Intuitively, one might assume that the frequency (and voltage) can be scaled adaptively over the cooling period. For every change in setting, dynamic voltage scaling (DVS) schemes stall for anywhere from 10 to 50μs to accommodate resynchronization of the clock's phase locked loop (PLL) [13]. Thus by selecting the correct frequency for throttling, we save on the expensive DVS stall period. On the other hand, if the frequency is not changed adaptively, an aggressive static frequency scaling may need to be employed to ensure reaching a safe temperature, which negatively impacts performance. One of the main criteria for DTM is speed of response. Our method is extremely fast, which takes 3ms on average and can be used in real time temperature control. In Section 3.1.3 we will elaborate on alternative methods and explain their runtime overhead, which makes them infeasible for DTM. Our specific contributions in this paper are summarized below. We

- Formulate the physical aware frequency selection problem for DTM in multi-core systems.
- Develop temperature models for frequency selection.
- Develop fast frequency selection mechanisms, which can be used for real time thermal management.
- Evaluate the frequency selection for different levels of aggressiveness and their effectiveness on real time thermal management. We also compare runtime of our technique in comparison to binary search based frequency selection.

The remainder of this paper is organized as follows. In Section 2 we elaborate on the physical aware frequency selection paradigm for DTM. In Section 3, we introduce the relevant thermal and power models used in this work. Our frequency selection algorithm is presented in Section 3.1. We discuss our experimental flow and results in Section 4. We conclude with a summary in Section 5.

2. PHYSICAL AWARE FREQUENCY SELECTION FOR DTM

We consider a simplified layout of a multi-core system as shown in Figure 1. Throughout the execution, the distribution of threads to different cores can lead to uneven amounts of activity in different cores. As a result, some cores exhibit hotspots. We must note that the task distribution on such systems will be performed primarily for performance and communication constraints (since interconnect delay will be a major component of performance). Therefore, for these systems, task distribution is unlikely to present a thermally good solution and localized activity will prevail leading to thermal emergencies. Nevertheless, thermal-aware task distribution will certainly be an important component of the thermal-aware system design paradigm and our techniques would co-exist with them in a comprehensive solution.

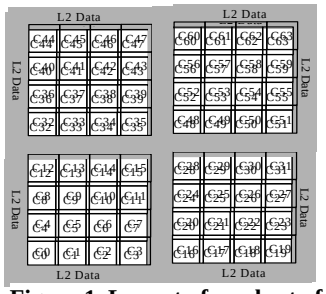

Figure 1. Layout of a subset of cores in a multi-core system.

The evolution of thermal effects with generations of multi-core systems can also be viewed from the following aspect. Powell et al. [7] assumed that the SMT cores in a CMP are thermally insulated by L2 cache blocks. The thermal effects of the adjacent blocks were not considered by Huang et al. [8], either. Kumar et al. [14] presented floorplans for 4, 8 and 16 core CMP. In architectures with fewer processor cores, each core tends to have a L2 cache placed nearby such that these relatively colder cache blocks provide significant thermal insulation. Instantiating multiple cores on a single chip improves performance per Watt efficiency [15] and it is predicted that we are headed into multi-core processor era [16]. As new architectures with increasing number of cores are developed, the arrangement of the cores in the layout is changing. For instance, eight synergistic processor elements are stacked next to each other in first generation Cell processor [17]. Intel IXP2800 has sixteen independent micro-engines arranged in an array [16]. Further, Intel is experimenting with tens of cores, potentially even hundreds of cores per single processor die [18]. The trends indicate that in future architectures there is a greater likelihood of the cores being placed physically adjacent without L2 caches.

Increasing physical interaction between cores in such large-scale systems becomes an important factor in shaping the thermal behavior of individual cores. If the physical sizes of the cores are sufficiently large, then the immediate neighbors play an important role in the thermal behavior of a core. For example in Figure 1, the neighbors of core 5 are cores 1, 4, 6, and 9. The lateral heat spreading into core 5 with respect to cores 0, 2, 8, and 10 (and other cores placed further away) is negligible and therefore they are not considered adjacent for the purpose of our temperature management. This provides an opportunity to deploy effective dynamic mechanisms to intervene in temperature emergency of a core by controlling the impact of its neighbors.

3. THERMAL AND POWER MODELS

In this work, we focus on a developing physical aware frequency scaling technique for DTM. First we present our relevant assumptions and models.

The dynamic power consumption of a single core due to switching activity is given by $P_{dynamic} = 0.5\alpha CV_{dd}^2 f$, where α is the switching activity of the core, V_{dd} and f are the operating voltage and frequency, respectively. Power consumed is dissipated as heat, which leads to a rise in temperature. The most commonly used technique for DTM by power reduction is Dynamic Voltage and Frequency Scaling (DVFS), where for each frequency reduction step the voltage is scaled to a level necessary to support the operating frequency. Our operating frequency and voltage levels for each core are assumed to be similar to Intel Pentium M Sonoma specification. We assumed that DVFS can be applied independently to each core. Similar assumptions about independent frequency and voltage control of each core can be found in existing literature [19]. The different frequency and corresponding voltage levels are shown in Table 1.

Table 1. Operating frequency and voltage levels.

Freq(GHz)	2.13	1.86	1.6	1.46	1.33	1.2	1.06	0.8
V_{dd} (V)	1.372	1.292	1.212	1.18	1.148	1.1	1.068	0.988
	F_{max}							F_{min}

In our framework the unsafe temperature condition T_O and the reset temperature T_R are set at 84.5℃ and 80.5℃ respectively. The target time interval τ for reducing hotspot temperature from T_O to T_R is to 200ms. The values of T_O, T_R and τ are based on the framework used by a commercial processor [6]. Our techniques can accept these values are parameters and operate under varying conditions. Our frequency selection mechanism requires the temperature reduction model (as we will discuss later in this section) and power consumption of the cores as its inputs. Isci et al. [20] proposed a technique for real time total power measurement using performance counter based per block power estimation. We assume that the power consumption can be estimated for each core using such performance counters and temperature of each core is available from thermal sensors.

We consider a multi-core architecture where cores are operating at maximum operating frequency (F_{max}) and corresponding V_{dd} as shown in Table 1. Let us assume that a core exhibits unsafe temperature condition. We illustrate the effect of different frequency (and voltage) scaling schemes on the temperature of this core. For example, let us assume that core 5 (as shown in Figure 1) is a hot core and exhibits unsafe temperature condition T_O. In Figure 2, we show the temperature of core 5 after cooling period τ for each of the frequency and voltage scaling steps from 1.86GHz to 0.8GHz. Each of the curves corresponds to an initial power value of the hot core P_h, where P_h is varied from 60W to 30W and the average power of its adjacent cores P_a is kept constant at 20W. It can be observed that the temperature reduction for the hot core at each step of frequency and voltage scaling shows a similar trend and only shifted by a constant factor for each initial power value P_h. The frequency has to be scaled to 1.06GHz at P_h equal to 60W to reduce temperature to 80.3°C within cooling period τ. In contrast, for P_h of 30W, scaling frequency to 1.6GHz is sufficient to reduce core temperature to T_R.

Another important aspect is the role of adjacent cores in temperature reduction of the hotspot. Figure 3 shows the temperature after cooling period τ by applying DVFS on hotspot (core 5) as the average power of adjacent cores P_a is varied from 14W to 28W. There is a reduction of 1.65°C of the hotspot temperature when the average adjacent core power is reduced from 28W to 14W. Power of the hotspot P_h is kept constant at 45W and each of the curves corresponds to a particular frequency (shown in Table 1) as P_a is varied. The curves show a similar trend for each frequency-scaling step from 1.86GHz to 0.8GHz. Figure 3 illustrates that temperature of a core is also a function of the physical interaction of the adjacent blocks.

Next we combine the results from Figure 2 and Figure 3 in Figure 4. Figure 4 shows the variation of temperature of the hotspot for different values of operating frequency for DVFS and average power of adjacent cores for a particular P_h after a time interval of τ. Surface plots of similar nature can be obtained for different values of P_h (based on the trends of the plots in Figure 2 and Figure 3). Thus, we have the temperature profile of the hotspot after τ as $T(F,P_a)\big|_{P_h=P_h'}$, $P_h' \in S_h$, where S_h is a set of values of P_h used for profiling. Due to the remarkably similar trends, surface plots can be obtained for P_h ($\notin S$) by interpolation or extrapolation such that we obtain the complete temperature profile after τ as $T(F,P_h,P_a)$. Temperature of the hotspot is expressed as function its own initial power and operating frequency for thermal management. It is also a function of average power of adjacent cores, which incorporates the physical awareness for DTM. This is in the shape of a surface in the three dimensional space, which we will refer as *thermal surface* in the remainder of our discussion.

We observe that the reduction of temperature after the cooling period τ is a smooth surface as a function of frequency scaling steps and average power of adjacent cores. The temperature of the hot core is monotonically decreasing with decreasing frequency and average power of the adjacent cores. Our frequency selection algorithm that intervenes to relieve thermal emergencies in a given core makes use of these observations.

Also, because of physical symmetry, this thermal surface is equally applicable for frequency selection of other cores having same number of thermally adjacent neighbors. Such a model is determined by the pre-defined cooling period τ required to reduce temperature from unsafe temperature T_O to reset value T_R. It is also dependent on the physical size of the cores, which is fixed for a multi-core architecture. The plots shown previously are for core 5 (in Figure 1). There will be two more types of surface plots, one for the cores with two adjacent neighbors (e.g. core 0) and one for the cores with three adjacent neighbors (e.g. core 4). The model needs to be generated from thermal simulation only once for an architecture and subsequently used by our mechanism for DTM.

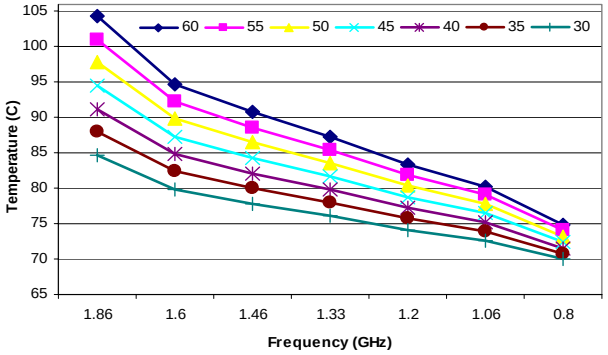

Figure 2. Reduction in temperature of core 5 at each frequency level for DVFS scheme.

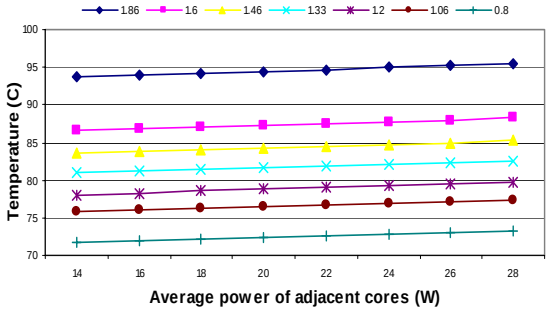

Figure 3. Effect of power and DVFS scheme applied to adjacent cores on the temperature of core 5, which has power consumption of 45W in this case.

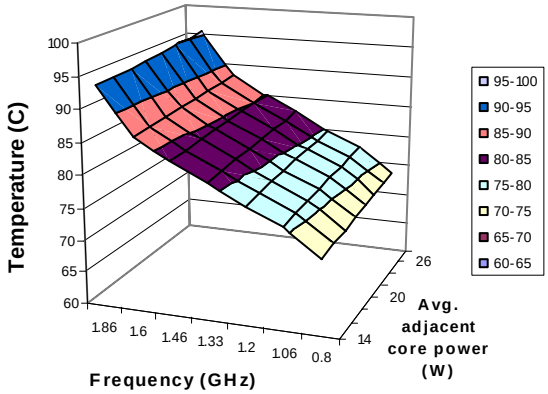

Figure 4. Temperature effect of DVFS scheme combined with the adjacent cores on core 5, which has 45W initial power.

3.1 An Algorithm for Frequency Selection

Having discussed our thermal and power models, next we describe how to determine optimized frequency scaling.

3.1.1 Problem Description

Determining the optimized frequency scaling for the hotspot, which is at an initial power of P_h' involves searching across the thermal surface for the pair of points *(F, P_a)* in the region less than or equal to the reset temperature i.e. $T(F,P_a)\big|_{P_h=P_h'} \leq T_R$.

Identifying the pair of points *(F, P_a)* will imply a new chosen

operating frequency F for the hot core and the required level of average power consumption P_a of the thermally adjacent neighbors. If the nature of the thermal emergency requires intervention on both the operating frequency of the hotspot and the power consumption of the thermally adjacent neighbors, then both F and P_a will take on new values. As a consequence we will determine the appropriate frequency scaling for the adjacent neighbors as well.

We illustrate this with an example in Figure 5. Let P_R be the average power over the adjacent cores. Along the thermal surface, all pairs of points *(F, P_a)* at the same P_R value form a curve (as in Figure 2). The feasible frequency scaling points are those on this curve for which temperature is less than T_R. Let the frequency F_R ($<F_{max}$) be required to reduce the hotspot's temperature to T_R within period τ, while keeping the average power over the adjacent cores unchanged at P_R. In this case, we determine our pair of points *(F, P_a)* as *(F_R, P_R)*. Although selection of F_R will relieve thermal emergency, this may not be the best selection in terms of performance of the hot core. In order to search for the best solution with minimal impact on the performance of the core we need to consider the physical interaction between adjacent cores and their respective timing criticality. This can yield a solution where instead of reducing operating frequency of the hot core aggressively, we can reach an alternative solution where physically adjacent cores help relieve thermal stress on the hotspot collectively. This decision will be weighted by the criticality of tasks executing on individual neighbors.

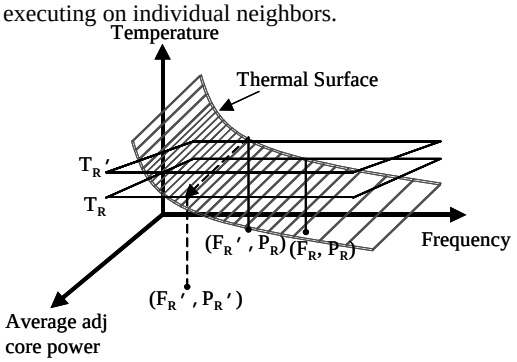

Figure 5. Illustration of frequency selection for reducing core temperature below reset value. $T_R' > T_R$, $F_R' > F_R$, $P_R > P_R'$.

Let us assume that a less aggressive frequency scaling F_R' for the hot core is chosen such that $F_R' > F_R$. The resulting reduction in temperature of the hotspot will be less; reaching T_R' such that $T_R' > T_R$. The thermal surface indicates that in order to reach safe temperature T_R, the average adjacent core power needs to be reduced from P_R to P_R'. In this case we determine our pair of points *(F, P_a)* as *(F_R', P_R')* where $F_R' > F_R$ and $P_R' < P_R$.

This involves applying power reduction techniques to the hot core as well as to its adjacent cores. The hotspot is still reduced from its unsafe temperature condition to its reset temperature. This enables operating the hot core in the throttling state at frequency F_R' as compared to F_R with lesser performance penalty. Frequencies higher than F_R' at which there is no reset temperature correspond to points on the surface plot (with feasible power values for adjacent cores) that are infeasible to the DVFS scheme. Of course by reducing power of the neighboring cores of the hotspot, we will incur a performance penalty to the neighbors. Such a decision is based on the criticality of the tasks executed by the neighboring cores and the hot core. In the next section, we will present two efficient algorithms to determine the amount of DVFS to be applied to the neighboring cores based on their criticality such that the safe temperature will be reached at the hotspot with minimum overall performance degradation. In summary, there are two observations that form the basis of our algorithm: (a) temperature of the hotspot is monotonically decreasing with decreasing frequency (F) and average power of the adjacent cores (P_a) and (b) the thermal surface describing the relationship between hotspot's power, temperature, and average power of the neighbor cores is bounded on the F and P_a axis. If there are no points for which temperature is at or below T_R at F_{min}, then DTM of the hot core cannot be performed by DVFS alone and prolonged throttling by clock gating has to be applied to the core. It also points to the opportunity of increasing the power of the adjacent cores of the hotspot beyond their present power P_R (with hot core reaching T_R after τ), which can be used for activity migration from the hot core to its neighbors.

3.1.2 Algorithm Description

Our frequency selection optimization works as follows. It takes as inputs the thermal surfaces that are generated for a multi-core architecture and the runtime power consumption of the cores. Once a thermal emergency for a core is detected, our algorithm determines the DVFS necessary to reduce the unsafe temperature of the hot core to reset temperature. The goal is to determine a pair of points *(F, P_a)* for a given P_h such that $T(F, P_h, P_a) \le T_R$ after τ.

The temperature model can be viewed as a function of frequency for fixed initial power values of the hot core and an average power across the adjacent cores. Such an equation can be expressed as $T(F)\big|_{P_h=P_h', P_a=P_a'}$ where hot core's power is fixed at P_h' and average adjacent core power is fixed at P_a'. We generate a set of such equations for different values of P_h and fixed P_a from Figure 2 by curve fitting. In order to adequately represent the thermal surface we also need the variation of temperature as the adjacent core power is changed for fixed hotspot's power and different frequencies, i.e., $T(P_a)\big|_{P_h=P_h', F=F'}$. This is obtained from Figure 3. The variation of temperature is linear with P_a and has the same gradient for different frequency scaling steps (and remains same for different initial power values of the hot core). We store the *gradient m* to create a set of equations $T(F, P_a)\big|_{P_h=P_h'}$ $\forall P_h' \in S_h$, where S_h is a set of hot core's initial power values. Equation for intermediate power values of the hotspot can be derived by interpolation or extrapolation. Thus the thermal surface $T(F, P_h, P_a)$ (discussed in Section 3) can be efficiently generated and stored.

Since frequency can be scaled to only certain discrete values, for every frequency scaling step the temperature of the hotspot after the cooling period can be determined from the thermal surface. Next, we determine two frequency points F_R and F_R' at the present average power of the adjacent cores $P_a = P_R$ such that $T_1(F_R', P_R)\big|_{P_h=P_h'} > T_R \ge T_2(F_R, P_R)\big|_{P_h=P_h'}$, where $F_R' > F_R$.

We propose two different types of frequency selection, which we refer as Aggressive Scaling and Criticality-based Scaling.

Aggressive scaling: Choosing frequency F_R for DVFS will reduce the hot core's temperature to (or below) T_R after τ. The solution is the frequency and average neighbor power coordinates *(F_R, P_R)*.

Criticality-based scaling: If we choose F_R', applying DVFS on the hot core alone will not suffice to attain T_R after time interval τ. However, this will enable us to explore a trade-off between the performance of the hot core and the frequency levels of its

neighbors. This scheme is physical-aware, where thermal management of the hotspot is performed by power reduction of the adjacent cores. At frequency point F_R', we determine the average power of the adjacent cores required to attain T_R after τ. Such a power value P_R' is obtained by $P_R^{'} = \frac{T_R - T_1}{m} + P_R$, where *m* is the gradient as discussed before. The temperature is monotonically decreasing with decreasing P_a. Therefore, $P_R' < P_R$.

The next task is to determine if the solution P_R' is valid and how to reduce the power of individual neighbors such that the average power is P_R'. There can be different techniques to reduce the power of the adjacent cores. In this work we are primarily focused on using DVFS scheme. Let us denote the power of i[th] adjacent core (as function of frequency) as $P_i(F)$ and $\frac{1}{n}\sum_{i=1}^{n} P_i(F_{\max}) = P_R$, where *n* is the number of adjacent cores. We want to determine a frequency of each core such that $\frac{1}{n}\sum_{i=1}^{n} P_i(F_i) \le P_R^{'}$. The solution (F_R', P_R') is infeasible if $\frac{1}{n}\sum_{i=1}^{n} P_i(F_{\min}) > P_R^{'}$. In this scenario, criticality based scaling reverts to aggressive scaling.

If there is a feasible solution, our algorithm next determines the frequency assignments of the cores adjacent to hot core. This scheme is applicable when the hot core is the most timing critical since all of its adjacent cores can be assigned reduced frequency and overall performance of the system will degrade. We attempt to prevent wide variation of power among the adjacent cores when their frequencies are scaled to obtain the target average power. The adjacent cores are sorted according to non-decreasing order of criticality. Starting from the core with lowest criticality, the power of each adjacent core is examined. If the present power of the core in i[th] iteration is more than $P_R^{'(i)}$ (average power requirement in the i[th] iteration), i.e., $P_i(F_{\max}) > P_R^{'(i)}$ then frequency is scaled to F_i such that $P_i(F_i) \le P_R^{'(i)}$. The reason we did not use the equality sign is to account for the discrete frequency scaling levels and hence, discrete power reduction levels of the adjacent cores. Then, the required average power of the remaining cores (to be used in (i+1)[th] iteration at which point we have processed i adjacent cores) is updated as $P_R^{'(i+1)} = \frac{1}{n-i}(P_R^{'(i)} \times (n-i+1) - P_i(F_i))$, where *n* is the number of adjacent cores.

The run time for aggressive scaling is constant time (O(1)) where the temperature is evaluated from the thermal surface for all the frequency points. The frequency scaling occurs in discrete steps and the number of such steps is only a few (equal to eight in Table 1). Criticality-based scaling takes *O*(*n*) time where *n* is the number of adjacent cores.

It might happen that during the period of applying frequency scaling for DTM, the application may go through a low Instruction Per Cycle (IPC) state with consequent power reduction of the cores. If the power of the hot core reduces as a result of such an application phase, our initial frequency selection remains unaltered. As a result, the DVFS step as determined earlier produces a conservative solution. On the other hand, if the power increases, then DVFS is pre-empted and the cores are allowed to run full throttle and a new frequency selection decision is carried out.

3.1.3 Alternative Approaches

In thermal simulators, a method to compute temperature solving a system of differential equation is presented [21] where ΔT_{tot} is the temperature contribution in time Δt due to power P_{tot} and thermal time constant R and C. Such an equation is shown below.

$$\Delta T_{tot} = \frac{P_{tot}\Delta t}{C} - \frac{T_i \Delta t}{R\,C}$$

The inverse of this equation may be applied to DTM where P_{tot} is unknown and ΔT_{tot} reduction in temperature is desired in time Δt. The form of the equation is shown below.

$$P_{tot} = C\frac{\Delta T_{tot}}{\Delta t} + \frac{T_i}{R}$$

The main problem with solving such an equation is that the nature of the temperature reduction curve is unknown and similarly, the shape of the gradient of temperature with respect to time is difficult to reconstruct. Such an equation is of type *integral equations of first kind* and can be solved by numerical methods. However solving this equation by numerical methods is not bounded in time.

Another alternative is to perform binary search of frequency scaling steps, simulate the temperature for each frequency scaling and determine the desired scaling to reach reset temperature. Thermal simulators solve a set of differential equations using numerical methods. Dynamic solutions using thermal simulation even for few elements (hot core and its neighbors) is expensive to be used for determining the frequency scaling in real time.

4. EXPERIMENTAL RESULTS

In the following sections, we first describe our experimental flow and then our results.

4.1 Experimental Setup

We considered an array of nine cores for our experimental purposes. Each core is of dimension 7mm × 7mm and similar to Alpha 21364 processor core without L2 cache. The first step is to create thermal surface models. We performed thermal simulation using HotSpot 3.0 [3] for a set of frequencies S_F (shown in Table 1), a set of average adjacent core powers ($P_a \in S_a$), and a set of initial power values of the hot core ($P_h \in S_h$) to create the thermal surface $T(F, P_h, P_a)$. Similar models are created for a corner core (having two thermally adjacent neighbors) and side core (having three thermally adjacent neighbors). The thermal surface and power consumption of the cores is input to our frequency selection algorithm. During our thermal simulation we fast-forward to an instant where a core exhibiting thermal emergency is detected.

Our physical aware frequency selection algorithm receives an interrupt to perform dynamic thermal management. In response, it performs frequency selection using three approaches. First approach is the aggressive frequency scaling. The second is the physical aware criticality based scaling. Due to frequency scaling, say to F_i (and corresponding operating voltage V_i) the hot core and its adjacent cores (for criticality based scaling) will operate at reduced power, which is determined as $P_{reduced} = P_{org} \times \frac{V_i^2 F_i}{V_{\max}^2 F_{\max}}$.

The cores are then simulated with the reduced power for a period of τ to validate that our technique was effective for thermal management. Finally, the third approach is to perform binary search and thermal simulation in each step to determine the frequency scaling for DTM. Although this method is most accurate, we show that the run time overhead makes it infeasible for real time thermal management.

4.2 Experimental Results

In this section we discuss our experimental results. The first set of results is for frequency levels obtained by aggressive and criticality-based scaling for different values of P_h. The hot core (ch) has four neighbors (c1-c4). Let us consider the hot core and adjacent cores 1, 2, 3, and 4 are in decreasing order of criticality. Figure 6 show the scaled frequencies determined by our algorithm for P_h equal to 45W and 30W respectively. The average power of the adjacent cores is 20W in this case. For P_h = 45W, the aggressive scaling determined the frequency scaling of the hot core to be 1.2 GHz while the frequency of the adjacent cores remains unchanged at 2.13GHz. The criticality-based frequency selection determined the frequency of hot core to be 1.33GHz. At the same time, the adjacent cores have to be operated at reduced power to help mitigate the hotspot. Adjacent core 1 has the highest criticality among the neighbors of the hot core. Its frequency is scaled to 1.86 GHz. As compared to cores 2, 3 and 4, which are scaled to 1.6GHz, from 2.13GHz. For P_h = 30W, hotspot is scaled to 1.6 GHz by the criticality-based scheme compared to 1.46 GHz by the aggressive scaling. The least critical core (core 4) is scaled to 1.86 GHz, while the other adjacent cores operate at 2.13 GHz. The criticality based scheme enables to operate the hot core, which also has the highest criticality in this case to operate at a higher frequency over that determined by aggressive scaling.

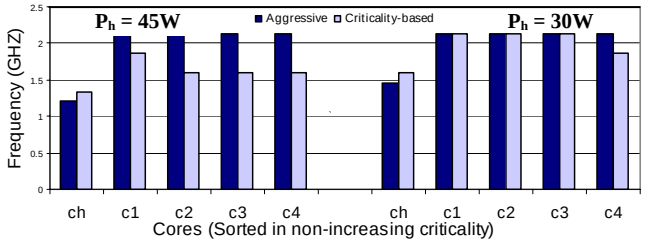

Figure 6. Frequency scaling determined by aggressive scaling and criticality based scaling schemes. Hot core has highest criticality. Adjacent cores 1, 2, 3 and 4 are in non-increasing order of criticality.

In Figure 7, we present a comparison between the operating frequencies selected for the hot core by the aggressive scaling and criticality based scaling. We show frequency selections made by these two schemes for different values of P_h. This shows that the criticality-based algorithm in almost all cases determines a higher operating frequency than that by aggressive scaling and leads to a 11.7% improvement in operating frequency on average.

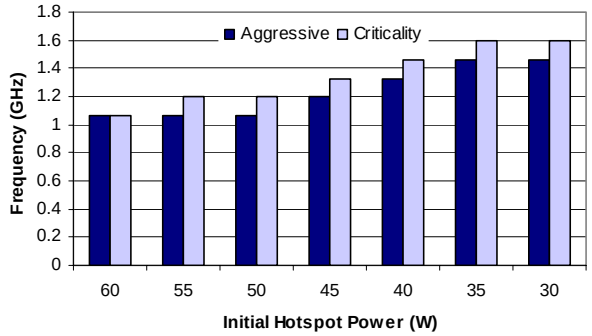

Figure 7. Comparison of the frequency scaling for aggressive and criticality based schemes.

Our experiments are run on 3.4GHz Intel Pentium 4 processor with 1GB memory. The runtime requirements for our aggressive and criticality based schemes to be 2 ms and 3 ms respectively. This shows that our approach can be indeed used for real time frequency selection. On the other hand an explicit binary search for the frequencies (and thermal simulation for each selection) takes 46.48 seconds, which is prohibitively expensive for a real time frequency selection mechanism.

5. CONCLUSIONS

We have introduced physical aware frequency selection for DVFS to assist dynamic thermal management in multi-core processors. At the same time we have minimized the stalls required for voltage scaling. Our simulation results show that the selected frequencies help in relieving the hot core from its thermal emergency.

Our physical aware criticality-based algorithm results in 11.7% faster operating frequencies compared to aggressive scaling. We also show that our technique is extremely fast and is suited for real time thermal management.

6. ACKNOWLEDGEMENTS

We would like to thank Dr. Robert Dick for helpful comments and suggestions. This research is supported in part by the NSF Career Award CNS-0546305 and the Northwestern University Alumnae Association Research Initiation Award.

7. REFERENCES

1. Mahajan, R. *Thermal management of CPUs: A perspective on trends, needs and opportunities*. in *Keynote presentation, THERMINIC-8*. 2002.
2. Borkar, S., *Design Challenges of Technology Scaling*. IEEE Micro, July-August 1999. **19**(4): p. 23--29.
3. Skadron, K., et al. *Temperature-Aware Microarchitecture*. in *International Symposium on Computer Architecture*. 2003.
4. Rotem, E., et al. *Analysis of Thermal Monitor features of the Intel® Pentium® M Processor*. in *Workshop on Temperature-aware Computer Systems*. 2004.
5. Sanchez, H., et al. *Thermal Management System for High Performance PowerPC Microprocessors*. in *IEEE Computer Society International Conference*. 1997.
6. Clabes, J., et al. *Design and Implementation of the POWER5 Microprocessor*. in *Design Automation Conference*. 2004.
7. Powell, M.D., et al. *Heat-and-Run: Leveraging SMT and CMP to Manage Power Density Through the Operating System*. in *International Conference on Architectural Support for Programming Languages and Operating Systems*. 2004.
8. Huang, W., et al. *A Framework for Dynamic Energy Efficiency and Temperature Management*. in *International Symposium on Microarchitecture*. 2000.
9. Li, Y., et al. *Evaluating the Thermal Efficiency of SMT and CMP Architectures*. in *IBM Watson Conference on Interaction between Architecture, Circuits, and Compilers*. 2004.
10. Donald, J. and M. Martonosi. *Temperature-Aware Design Issues for SMT and CMP Architectures*. in *Workshop on Complexity-Effective Design* 2004.
11. Brooks, D. and M. Martonosi. *Dynamic Thermal Management for High-Performance Microprocessors*. in *International Symposium on High-Performance Computer Architecture*. 2001.
12. Klaiber, A. *The Technology Behind Crusoe Processors. Whitepaper*. 2000
13. Skadron, K., et al., *Temperature-Aware Microarchitecture: Extended Discussion and Results*, in *University of Virginia Dept. of Computer Science Technical Report CS-2003-08*. April 2003.
14. Kumar, R., et al. *Interconnections in Multi-Core Architectures: Understanding Mechanisms, Overheads and Scaling*. in *International Symposium on Computer Architecture*. 2005.
15. Hofstee, H.P. *Power Efficient Processor Architecture and The Cell Processor*. in *Symposium on High Performance Computer Architecture*. 2005.
16. Borkar, S., et al. *Platform 2015: Intel Processor and Platform Evolution for the Next Decade. Whitepaper* 2005
17. Pham, D., et al. *The Design and Implementation of a First Generation Cell Processor*. in *International Solid-State Circuits Conference*. 2005.
18. Rattner, J.R. *Keynote at the Intel Developer Conference* 2005
19. Juang, P., et al. *Coordinated, Distributed, Formal Energy Management of Chip Multiprocessors*. in *International Symposium on Low Power Electronics and Systems* 2005.
20. Isci, C. and M. Martonosi. *Runtime Power Monitoring in High-End Processors: Methodology and Empirical Data* in *International Symposium on Microarchitecture*. 2003.
21. Skadron, K., et al. *Control-Theoretic Techniques and Thermal-RC Modeling for Accurate and Localized Dynamic Thermal Management*. in *International Symposium on High-Performance Computer Architecture*. 2002.