Frequency Planning for Multi-Core Processors Under Thermal Constraints

Michael Kadin
Division of Engineering
Brown University
Providence, RI 02912
michael_kadin@brown.edu

Sherief Reda
Division of Engineering
Brown University
Providence, RI 02912
sherief_reda@brown.edu

ABSTRACT

The objectives of this paper are (1) to develop a frequency planning methodology that maximizes the total performance of multi-core processors and that limits their maximum temperature as specified by the design constraints; and (2) to establish the implications of technology scaling on the performance limits of multi-core processors. Given the intricate designs and workloads of multi or many-core processors, it is computationally exhaustive to develop models that accurately calculate the temperature and performance of a given processor under various operating conditions. To abstract the underlying design complexity, we propose the use of supervised machine learning techniques to develop versatile models that capture the thermal characterization of multi-core processors under various input conditions and workloads. We then use the developed models to create a framework where various design constraints and objectives are expressed and solved using combinatorial optimization techniques. Using established power modeling and thermal simulation tools, we show that it is possible to boost the performance of multi-core processors by up to 11.4% at no impact to the maximum temperature.

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

1. INTRODUCTION

Elevated chip temperatures arising from increased power densities associated with sub-100 nm technologies are significant limiters to potential performance improvements from CMOS technology. Besides hindering frequency increase, high temperatures increase leakage current, slow down transistors and interconnects, and can potentially damage circuits by accelerating their aging. Dynamic thermal management (DTM) techniques adjust the thermal status of the chip at runtime to prevent any damage to it from high temperatures. By reading in the thermal state of the chip (using performance counters or temperature sensors), a thermal management scheme can respond to thermal emergencies [5, 3, 8]. Dynamic frequency and voltage scaling of multi-core processors can provide leverage to control the performance, temperature and power consumption of multi-core processors [10, 9].

To keep the maximum temperature within a reasonable range (e.g., ≤ 85℃), it is necessary to bound the operating frequency of the individual cores, which in turn caps the single-thread performance as well as the total physical performance. The objective of this paper is to put forward a design methodology that calculates the frequency plan, i.e., the frequency of each individual core, of a multi-core processors under thermal constraints. Independent control of the operating parameters of individual cores is readily feasible [7]; for example, each core in AMD's Phenom quad-core processor has its own phase locked loop and clock distribution network and power grid, which allows each core to control its operating frequency [2]. Towards developing a design methodology that derives optimal frequency plans, we propose a number of novel techniques. The contributions of this paper are as follows.

1. We propose the use of supervised machine learning techniques to abstract the underlying design complexity and to develop versatile models that capture the thermal characterization of multi-core processors under various input conditions and workloads.
2. We use the learned models within a combinatorial optimization framework to calculate optimal frequency plans that maximize the total performance of multi-core processors under thermal constraints.
3. We study the thermally-constrained physical performance of different multi-core configurations at a number of semiconductor technology nodes. Our results provide "first-order" performance scaling trends for multi-core processors till the 22 nm technology node.
4. We show that it is possible to devise frequency plans that boost the total performance of multi-core processors by up to 11.5% at no impact to the maximum temperature. Furthermore, the results reveal the interplay between the spatial location of a core and its frequency.

This paper is organized as follows. In Section 2 we discuss the details of our proposed methodology. Section 3 provides a comprehensive set of experimental results for our techniques and gives the main conclusions of our work.

ISLPED'08, August 11–13, 2008, Bangalore, India..

2. PROPOSED METHODOLOGY

The objective of this work is to devise optimal frequency plans for multi-core processors under thermal constraints. We propose the following three-step methodology.

1. **Thermal Characterization (Subsection 2.1).** The first step is to exercise the processor at various operating frequencies, and measure the resultant temperatures at every unit of the processor. This exercising can be carried out either within a simulation environment or in a physical system where the temperatures are measured (e.g., using an IR camera or die-integrated temperature sensors).
2. **Modeling and Validation Using Supervised Machine Learning Techniques (Subsection 2.2).** Given the input and output measurements from Step 1, apply supervised machine learning techniques to construct a model that best describes the measured results.
3. **Optimal Frequency Planning (Subsection 2.3).** In this step, the desired performance-maximization objective and temperature constraints are expressed within the framework of the model produced from Step 2. The model is solved using standard combinatorial optimization techniques to determine the optimal frequency plan.

2.1 Thermal Characterization

The first step is to thermally characterize the behavior of a multi-core processor as a function of its operating frequencies. As mentioned earlier, thermal characterization can be achieved through either physical measurements, or through a simulation tool chain. In this paper, we will focus on thermal characterization using tool chains.

To calculate the junction temperature vector of all functional units from the given frequencies, a tool chain is setup as given in Figure 1. The tool chain is also given as inputs the processor's description, its layout organization as well as potential workload applications. Due to the dependence of leakage on temperature, the tool chain is setup in an iterative fashion, where the temperature measurements are fed back to the leakage power calculator to update its results. The flow is typically iterated a few times until stable power and temperature values are attained. In our tool chain, we use HotSpot 4.0 [11] as the thermal simulator, as well as PTScalar 1.0 [6] and CACTI 5.0 [12] for power calculations. We use the Alpha EV6 processor as our baseline processor. For workloads, we use 8 benchmarks from the SPEC2000 suite. We use 4 integer benchmarks: gcc, bzip, mcf, and twolf, and 4 floating point benchmarks: ammp, equake, lucas, and mesa.

Tool chains are useful in characterizing the operating temperatures of a given design under various operating frequencies. However, in the case of processor designs with many cores, they can be ineffective in determining the optimal operating frequencies of the individual cores. The exponential increase in search space, as the number of cores increases, renders the explicit enumeration and evaluation of every potential choice for the operating frequencies computationally infeasible. The methods proposed in the next two subsections extend traditional tool chains by abstracting their behavior in a mathematical formulation that can be used to quickly arrive at the optimal frequency plans without the need for explicit search.

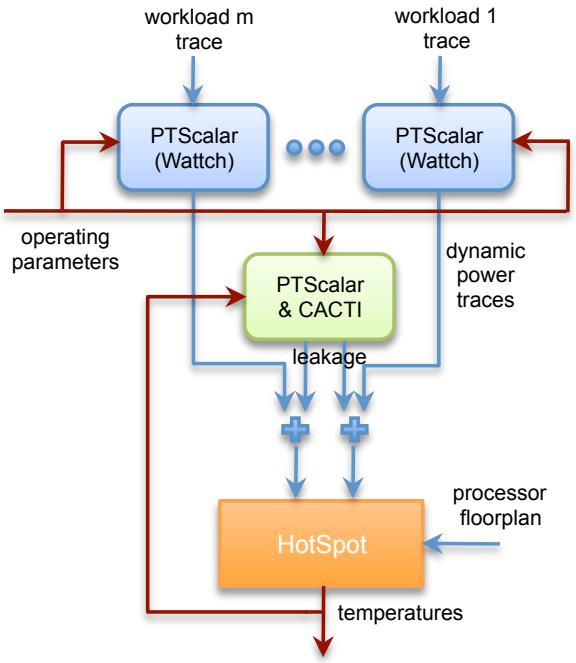

Figure 1: A tool chain to calculate the junction temperature of various processor units as a function of input frequencies.

2.2 Modeling and Validation

Towards proposing a reasonable machine learning (ML) model, we utilize the well-known duality between RC circuits and thermal systems. In thermal RC models, temperature is analogous to voltage, and current is analogous to heat transfer. Resistances represent paths of heat transfer, current sources represent power dissipation, and voltage sources represent constant temperature sources. Thermal capacitors, represent the ability to store heat, and are included in transient analysis (as opposed to steady-state) to model the time dependant changes in temperature. Consider a very simple system with two "generic" functional units as shown in Figure 2. In the figure, unit 0 connects to unit 1 via a thermal resistance R_a. Both units are connected to the package through additional thermal resistances R_0 and R_1. The package is then connected to the ambient air through R_p. Each unit also generates its own power, P_0 and P_1, which is modeled as a thermal current source. In this work, we model the steady-state junction temperature. Therefore, thermal capacitors are fully "charged" and can be omitted. Solving Kirchhoff's Current Law for the above system yields the following results for unit 0:

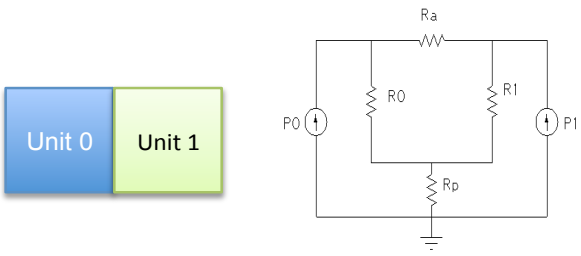

Figure 2: A simple example of heat transfer models.

$$t_0 = \frac{R_aR_p + R_aR_0 + R_1R_0 + R_pR_0 + R_pR_1}{R_1 + R_0 + R_a}p_0 + \frac{R_aR_p + R_pR_0 + R_1R_0 + R_pR_1}{R_1 + R_0 + R_a}p_1,$$

which can be succinctly described by $t_0 = \lambda_{0,0}p_0 + \lambda_{0,1}p_1$. Our objective is to avoid explicit calculation of the values of λs and to rather learn their values given the simulation results of tools such as HotSpot or physical-based measurements. Thus for a generic N unit system, we write the temperature at some unit i as

$$t_i = \sum_{j=1}^{N} \lambda_{i,j} p_j \quad (1)$$

In addition, we propose that the total power (static and dynamic) of a unit can be approximated as a linear function of its frequency. Therefore, if we assume a constant supply voltage, the power of a given unit j can be approximated by $p_j = a_j f_j + b_j$, where a_j models the direct impact of frequency on dynamic power and its indirect impact on leakage, and b_j gives the nominal leakage power. Substituting the power model in Equation 1 yields

$$t_i = \sum_{j=1}^{N} \lambda_{i,j}(a_j f_j + b_j) = \sum_{j=1}^{N} \lambda_{i,j} a_j f_j + \sum_{j=1}^{N} \lambda_{i,j} b_j \quad (2)$$

$$= \sum_{j=1}^{N} \pi_{i,j} f_j + \theta_i \quad (3)$$

where $\pi_{i,j}$ and θ_i are the model parameters. Thus for all N units, we find

$$\begin{pmatrix} t_1 \\ t_2 \\ \cdot \\ \cdot \\ \cdot \\ t_N \end{pmatrix} = \begin{pmatrix} \pi_{1,1} & \cdots & \pi_{1,N} \\ \pi_{2,1} & & \pi_{2,N} \\ \cdots & & \\ \pi_{N,1} & \cdots & \pi_{N,N} \end{pmatrix} \cdot \begin{pmatrix} f_1 \\ f_2 \\ \cdot \\ \cdot \\ f_N \end{pmatrix} + \begin{pmatrix} \theta_1 \\ \theta_2 \\ \cdot \\ \cdot \\ \theta_N \end{pmatrix} \quad (4)$$

which can be written succinctly in matrix notation as $T = \Pi F + \Theta$. Consider the case of a 16 core system based on the Alpha EV6 processor core, where each core consists of 17 individual functional units, and the cache consists of four big shared blocks. In this case the total number of units N in the processor is equal to $17 \times 16 + 4 = 276$. Since units that belong to the same core share the same operating parameters, the model parameters, defined by Π and Θ, have a total of $276 \times 16 + 276 = 4692$ parameters to be learned.

Learning the Model Parameters. Given the ML model, the objective is to learn the ML model parameters, Π and Θ, that give temperature results $T = \Pi F_i + \Theta$ that are closest to the observed temperature measurements T^{obs} and minimize the average squared error between some k observed and calculated temperatures, i.e., $\frac{1}{k}\Sigma_{i=1}^{k}(\Pi F_i + \Theta - T_i^{obs})^2$. We propose to use an iterative learning process that can be described as follows. First, apply a Monte Carlo simulation to thermally characterize a given multi-core processor using random sets of operating frequencies. Second, use the operating vectors together with the observed temperatures to find the solution to min $\frac{1}{k}\Sigma_{i=1}^{k}(\Pi F_i + \Theta - T_i^{obs})^2$ using robust multiple regression techniques. Third, the error in model estimation is calculated, and if the error is higher then a specified threshold then the learning process is iterated with a new round of Monte Carlo simulation until the error in estimation becomes less than some specified threshold. Note that the proposed learning approach provides a natural way to determine how much characterization is sufficient.

2.3 Optimal Frequency Planning

The developed model in the previous subsection enables the calculation of frequency plans under thermal constraints. If the maximum allowed temperature at any unit in the die is denoted by $T_{\max}$, then we can express the *feasible* set of frequency plans by $\Pi F + \Theta \leq T_{\max}$. The previous inequality defines a polytope that gives the range of frequencies where the maximum temperature constraint is not violated. From the set of all feasible frequency plans, we desire to find the optimal one. We consider two cases.

A. Standard Frequency Planning. In this traditional case, all cores run at the same frequency, and and thus the frequency vector is equal to $F = (f\ f\ \cdots\ f)^T$ and the objective function of is max f such $\Pi F + \Theta \leq T_{\max}$.

B. Optimal Frequency Planning. To find the individual core frequencies that maximize the total system physical performance, we set up an objective functional, max $\sum_i f_i$ such that $\Pi F + \Theta \leq T_{\max}$. Note that all units that belong to the same core share the same frequency. The sum of all frequencies is a reasonable metric to measure the total performance of a processor, and it will give the true throughput when the workloads are independent.

Both standard and optimal planning can be solved using Linear Programming (LP) techniques. In the LP, it is possible to include an upper bound on the individual core frequencies to avoid timing errors. The maximum frequency limit is calculated based on the technology and design used.

3. EXPERIMENTAL RESULTS

We use our proposed methodology to calculate optimal frequency plans for multi-core processors under thermal constraints. We use the tool chain given in Figure 1, which we described earlier in Subsection 2.1. We apply our method to six technology nodes 190 nm (1 core), 90 nm (2 cores), 65 nm (4 cores), 45 nm (8 cores), 32 nm (16 cores) and 22 nm (32 cores). For all technology nodes we assume a die size of 1.6×1.6 cm^2 and adjust the layout of the Alpha EV6 processor appropriately depending on the number of associated cores. To provide reasonable estimates of the key parameters at these nodes, we use the International Technology Roadmap for Semiconductors' (ITRS) values [1] (2005 edition with 2006 update), which we provide in Table 1.

We first consider standard frequency planning (Subsec-

Technology Node (nm)	Bulk silicon			FD SOI		
	130	90	65	45	32	22
Number of Cores	1	2	4	8	16	32
Supply V_{DD}	1.3	1.2	1.1	1.0	0.9	0.8
Leakage / μm width	0.01	0.05	0.20	0.22	0.29	0.37
Cap. (fF) / μm width	1.2	9.9	6.9	7.4	6.2	5.3

Table 1: Summary of power scaling results according to ITRS predictions.

plan	row 1 row 2	technology Node (nm) number of cores	130 nm 1	90 nm 2	65 nm 4	45 nm 8	32 nm 16	22 nm 32
standard	row 3	total performance	1.00	1.87	3.71	6.18	11.99	21.49
planning	row 4	$f_{\max}$ (GHz)	2.35	2.20	2.18	1.81	1.76	1.57
optimal planning	row 5	total performance	1.00	1.89	3.81	6.29	12.96	23.94
standard vs. optimal	row 6	improvement (%)	0.00%	0.01%	0.48%	1.87%	8.11%	11.39%

Table 2: Physical performance results multi-core systems for the standard and optimal planning. Performance is the total number of cycles executed per second by the processor normalized to the case of 1 core.

tion 2.3.A). We calculate the maximum frequency that can be used by all cores such that the junction temperature at any point in the processor does not exceed 85 ℃. In row 3 of Table 2, we report the total performance as measured by the total number of cycles executed per second normalized to the single-core processor frequency at the 130 nm node. The maximum core frequency is reported in row 4. The decline of the maximum frequency with every technology generation reduces the single-thread performance, which leads to the first main conclusion of this paper.

Conclusion 1. The utility value of an individual core depreciates with each new technology node.

We utilize our methodology to devise an optimal frequency plan that boosts the total performance (Subsection 2.3.B). Row 5 gives the total performance as measured by the total number of cycles executed per second normalized to the 130 nm processor. Row 6 gives the improvement in performance over standard planning. Comparing the standard and optimal plans leads to our second main conclusion.

Conclusion 2. Optimal frequency planning significantly improves the total performance compared to standard planning. Furthermore, the improvement increases with every technology node peaking up to 11.4% at the 22 nm node.

We also bar-plot the individual core frequencies for the 16 core system in Figure 3, where the 16 cores are organized a 4×4 layout grid. The bar plot shows that our devised frequency plans lead to a maximization of the frequencies in the outer cores compared to the inner cores, which leads to the third main conclusion.

Conclusion 3. Our results intuitively point out that in many-core systems, the inner located cores should generally run at slower frequencies than the outer cores as their lateral heat transfer must go through the outer cores, which consequently can increase the outer cores' temperature and end up slowing down the entire system.

To validate our results, we used the frequencies computed from the solution of the LP as inputs to the tool chain of Figure 1 to accurately re-calculate the temperatures. We compared the obtained temperatures against our model setting which is 85℃. We have found the average error is 0.18% and the maximum error is 1.68%. The results confirm that our methodology produces solutions that accurately satisfy the maximum temperature constraint. Finding the optimal solutions for the various technology nodes took a runtime of less than 0.01 seconds to compute with Matlab's LP solver.

Our future work will focuses on the following extensions: (i) adaptive tuning of the frequency plan within a dynamic thermal-management environment, and (ii) simultaneous frequency and voltage planning.

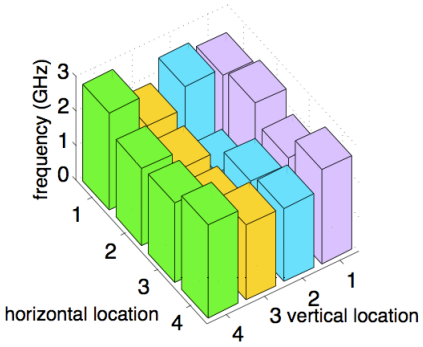

Figure 3: The frequency plan for a 16-core processor organized in a 4×4 layout.

4. REFERENCES

[1] "International Technology Roadmap for Semiconductors." [Online]. Available: http://public.itrs.net

[2] J. Boyd, "Native Quad Core Joins X86 Fray," in *EETimes.com*, 10/01/2007.

[3] D. Brooks and M. Martonosi, "Dynamic Thermal Management for High-Performance Microprocessors," in *HPCA*, 2001, pp. 171–182.

[4] D. Brooks, V. Tiwari, and M. Martonosi, "Wattch: A Framework for Architectural-Level Power Analysis and Optimizations," in *ISCA*, 2000, pp. 83–94.

[5] J. Donald and M. Martonosi, " Techniques for Multicore Thermal Management: Classification and New Exploration," in *ISCA*, 2006, pp. 78–88.

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

[7] G. Magklis, G. Semeraro, D. H. Albonesi, S. G. Dropsho, S. Dwarkadas, and M. L. Scott, "Dynamic Frequency and Voltage Scaling for A Multiple-Clock-Domain Microprocessor," *IEEE Micro*, vol. 23(6), pp. 62–68, 2003.

[8] R. Mukherjee and S. O. Memik, "Physical Aware Frequency Selection for Dynamic Thermal Management in Multi-Core Systems," in *ICCAD*, 2006, pp. 547–552.

[9] S. Murali, A. Mutapcic, D. Atienza, R. Gupta, S. Boyd, and G. D. Micheli, "Temperature-Aware Processor Frequency Assignment for MPSoCs Using Convext Optimization," in *CODES+ISSS*, 2007, pp. 111–116.

[10] R. Rao, S. Vrudhula, and C. Chakrabarti, "Throughput of Multi-Core Processors Under Thermal Constraints," in *ISLPED*, 2007, pp. 201–206.

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

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