A Power Capping Controller for Multicore Processors

N. Almoosa, W. Song, Y. Wardi and S. Yalamanchili
School of Electrical and Computer Engineering
Georgia Institute of Technology
{nawaf, wjhsong}@gatech.edu, {ywardi, sudha} @ece.gatech.edu

***Abstract*— This paper presents an online controller for tracking power-budgets in multicore processors using dynamic voltage-frequency scaling. The proposed control law comprises an integral controller whose gain is adjusted online based on the derivative of the power-frequency relationship. The control law is designed to achieve rapid settling time, and its tracking property is formally proven. Importantly, the controller design does not require off-line analysis of application workloads making it feasible for emerging heterogeneous and asymmetric multicore processors. Simulation results are presented for controlling power dissipation in multiple cores of an asymmetric multicore processor. Each core is i) equipped with the controller, ii) assigned a power budget, and iii) operates independently in tracking to its power budget. We use a cycle-level multicore simulator driven by traces from SPEC2006 benchmarks demonstrating that the proposed algorithm achieves a faster settling time than examples of a static setting of the controller gain.**

I. Introduction

The effective and efficient control of power and energy has become central to the design and management of modern computer systems. It is no longer just the domain of embedded and mobile devices but is as important for enterprise-class data centers and internet-server farms that can consume tens of megawatts of power [1]. The prevailing design methodology has been to design processors and systems based on peak power dissipation which is based on worst-case application workloads. This approach has a number of undesirable consequences. For example, a data center's cooling capacity and peak processor power dissipation together determine the density of servers that can be placed within the facility. However, rarely are all servers operating at peak power or utilization, and consequently the center is over-provisioned with a lower average performance per square foot. Similarly, the packaging cost of a multicore processor is determined by the target peak power dissipation. Higher package costs incurred by high peak power dissipation targets increases the cost of the processor even though peak loads may rarely occur in practice. Furthermore, as power densities increase at future technology nodes we will see an increase in the relative inefficiency of designing systems based on worst-case workload and peak power dissipation. This trend is unsustainable.

Researchers have observed that reducing the peak power dissipation design target leads to relatively little drop in execution performance reflecting the non-linear relationship between power and execution performance. However, suitable controls must be in place to prevent a processor from exceeding the power dissipation target in the unlikely event of a workload spike that would increase the power dissipation beyond this design target leading to disruptive failures. Consequently, to improve the cost-effectiveness of the systems several on-line control techniques have emerged that adjust system parameters to limit power consumption [2], [3], [4] (see also references therein). These techniques are based on dynamic scaling of the voltage and/or clock frequency for controlling the power dissipated by a processor in order to limit it to a certain value called the *power cap*. For example, the authors in [4] proposed a feedback controller for capping the power of voltage islands in chip multiprocessors, whose parameters are derived based on extensive off-line system analysis under various workload conditions.

Similarly, for data-center applications [2] proposed a proportional controller based on a linear system-model. To the best of our knowledge, [2] contains the first analysis of stability, convergence rate, and robustness of a control system for a blade server level power capping controller. The objective in that paper is to regulate the power dissipated by blade servers in order to have them track given reference values. The plant, comprised of the frequency-power relationship, is assumed to be linear and memoryless, and this assumption is backed by simulation tests for some specific workloads under a variety of conditions. The controller, relating the error (difference between the power-reference value and the actual power) to the frequency, is an integrator scaled by the reciprocal of the plants gain.[1] If the plants gain is known exactly, the control algorithm will converge in a single step, namely the power will be equal to its reference value in a single iteration. However, the gain may not be known exactly, and therefore the paper carries out convergence and robustness analysis to establish asymptotic convergence of the control law and stability margins in terms of bounds on the modeling error.

The work reported here has been motivated by [2], but it considers a more general model where the frequency-power relationship is nonlinear and the core architecture may no longer be homogeneous. Thus, we extend the results in [2] in the following ways - (i) The frequency-power characteristics are convex and not linear, and (ii) the controller is an integrator with a variable gain as opposed to a constant gain.

[1]Reference [2] defines the controller as the relationship between the incremental error between consecutive samples and the corresponding incremental power measures; as such the controller is proportional (linear). However, viewed as the relationship between error and power, the same controller is actually an integrator.

We then consider the particular case of cubic polynomials for the plant, supported by physical modeling, and show that the controller's gains are computable in an adaptive fashion based only on measurements but not on any off-line analysis. This is a significant advantage since unlike previous techniques, this controller can be packaged as part of a multicore blade server without any a priori knowledge of the applications that are to be executed. Finally, as in [2] we analyze the asymptotic convergence of the control algorithm as well as its stability and robustness, but as we shall see, this analysis is quite different from the one in [2]. Finally, we demonstrate the efficacy of our control law for benchmark programs executing on multicore processors.

The rest of the paper is organized as follows. Section II presents some of the main challenges arising in power control in multicore systems. Section III describes the proposed control law in an abstract setting and analyzes its asymptotic convergence. Section IV describes the specific control problem that is addressed in this paper, and Section V presents simulation results. Finally, Section VI concludes the paper.

II. System Model and Challenge

The target application domain is that of multicore processors. An example is a four core processor where each core has L1 instruction and data caches and a private L2 cache. Cores communicate with each other and with memory controllers and I/O devices through an on-chip network. However, we are concerned with an emerging class of multicore processors that are asymmetric in the designs of the cores, i.e., not all cores on the chip are of the same design [5], [6], [7].

In the example we evaluate in Section V, there are two types of cores. A complex out-of-order (OOO) core which employs aggressive pipelining and speculation to increase the average number of instructions executed per clock cycle (IPC). The second type of core is a simple in-order (IO) core where instructions are executed and retired in order. While multiple instructions can be issued in parallel, they are executed and retired in-order leading to lower average IPC. IO cores consume significantly less power than OOO cores. The emergence of asymmetric multicore processors reflect an architectural approach to constraining the rapidly growing power densities of future processors. OOO cores can provide high performance for critical single thread or serial segments of code (at higher power) while the IO cores can provide significantly better energy and power efficiency by executing parallel segments of code.

There are two issues when applying contemporary control techniques. First, a single controller for all types of cores is ineffective in such an architecture since the consequences of changing the voltage-frequency setting is very different for different types of cores. The natural choice is to have each core and its private caches be separately controlled. This implies that each core is in a separate voltage island which is quite common. For example, Intel's 48 core single chip cloud computer has 8 voltage domains and 28 frequency domains [8]. However, even then we have the second issue - contemporary power capping controller designs that rely on extensive offline analysis of applications to determine parameters of the model present practical impediments for deployment. The off-line analysis must be completed for all combinations of core types and applications. Different core-application combinations will most likely lead to controller designs with different gain parameters and convergence properties. This leads to the need for either i) restricting the cores on which specific applications can be executed or ii) the ability to change the controller gain as application threads are scheduled on different cores. The former defeats the purpose of having asymmetric multicore processors. The latter is an ad-hoc solution that is still limited since all gain values must be statically known.

We believe that the approach and design proposed here is superior in that our controller does not rely on extensive off-line analysis. We would observe that the controller design is based on fundamental frequency-power relationships that is experienced across all application and core combinations. Thus, the controller can be an integral part of the multicore design and be applicable across a wide range of applications and core types. The following sections provide the details of our approach.

III. Control Law

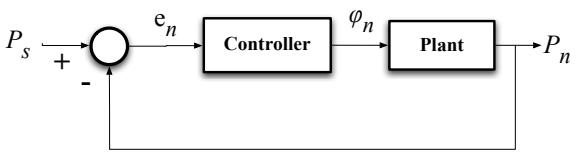


Fig. 1. Power Control System

Consider the discrete-time scalar system shown in Figure 1, where the plant is modeled as a memoryless, time-varying nonlinear system of the form $P = g_n(\phi)$; n denotes (discrete) time and $g_n : R \to R$ is the function defining the system at time n. In the next section ϕ_n and P_n will denote frequency and power, and this is the reason we are using the unusual notation for the control variable and the output signal, respectively. Suppose that the functions g_n, $n = 1, 2 \ldots$, have a common domain, $I := [\phi_{min}, \phi_{max}]$, where the following assumption is in force.

Assumption 1: Each one of the functions g_n is continuously differentiable, convex, and monotone-increasing throughout I. Furthermore, there exist constants $\gamma_1 > 0$ and $\gamma_2 < \infty$ such that, for every $n = 1, 2, \ldots$, $g_n^{'}(\phi_{min}) \geq \gamma_1$, and $g_n^{'}(\phi_{max}) \leq \gamma_2$ ('prime' denotes derivative with respect to ϕ).

We implicitly assume that every point ϕ mentioned in the sequel is contained in I.

Let P_s be a given reference input, and suppose that the purpose of the controller is to regulate the output in the sense that $\limsup_{n\to\infty} P_n$ and $\liminf_{n\to\infty} P_n$ are close to P_s within a certain tolerance. To this end we use an integral

controller of the form

$$\phi_n \;=\; \phi_{n-1} + K_n e_{n-1} \tag{1}$$

for a suitable gain $K_n > 0$, where we recall from the plant definition that

$$P_n \;=\; g_n(\phi_n), \tag{2}$$

and it is evident from Figure 1 that

$$e_n \;=\; P_s - P_n, \tag{3}$$

for all $n = 1, \ldots$. Thus, once the gains K_n, $n = 1, 2, \ldots$ are specified, Equations (1)-(3) define the closed-loop system in a recursive manner.

Suppose that at time n the function $g_n(\cdot)$ is known, we have a measurement of the control signal ϕ_{n-1}, and are able to compute the derivative term $g_n^{'}(\phi_{n-1})$. We now assume that the latter computation is exact, and later will consider approximations to it. We set the gain K_n to the following value,

$$K_n \;=\; \frac{1}{g_n^{'}(\phi_{n-1})}. \tag{4}$$

We point out that if the plant is time invariant, namely $g_n(\cdot) = g(\cdot)$ for some function $g : R \to R$ satisfying Assumption 1, then the recursive computation of e_n, defined by Equations (1) - (4), effectively is Newton's method for finding a zero of the equation $e = P_s - g(\phi) = 0$. In this case we have the following result.

Proposition 1: Suppose that the plant is time invariant. Then there exists a positive constant $\beta < 1$ such that, for every $n = 1, 2, \ldots$,

1) If $e_{n-1} \geq 0$ then $e_n \leq 0$.
2) If $e_{n-1} \leq 0$ then

$$\beta e_{n-1} \;\leq\; e_n \;\leq\; 0. \tag{5}$$

This result is a special case of Proposition 2, below, concerning time-varying systems.

As a corollary, it follows that the output tracks the reference input, since $\lim_{n\to\infty} e_n = 0$ and hence $\lim_{n\to\infty} P_n = P_s$. Moreover, this convergence is exponential in the sense that $|e_n| \leq A\beta^n$ for some $A > 0$ and $\beta \in (0, 1)$.

Consider now the time-varying case, where the closed-loop system is defined via Equations (1) - (4). The error term e_n satisfies the following inequalities.

Proposition 2: There exists a positive constant $\beta < 1$ such that, for every $n = 1, 2, \ldots$,

1) If $e_{n-1} \geq 0$, then

$$e_n \;\leq\; g_{n-1}(\phi_{n-1}) - g_n(\phi_{n-1}). \tag{6}$$

2) If $e_{n-1} \leq 0$, then

$$\begin{aligned} &\beta e_{n-1} + \big(g_{n-1}(\phi_{n-1}) - g_n(\phi_{n-1})\big) \\ &\leq\; e_n \;\leq\; g_{n-1}(\phi_{n-1}) - g_n(\phi_{n-1}). \end{aligned} \tag{7}$$

Proof: Consider a differentiable convex function $g : R \to R$. By the definition of convexity, for every $x \in R$ and $\Delta x \geq 0$, the following inequalities are in force:

$$g^{'}(x)\Delta x \;\leq\; g(x + \Delta x) - g(x) \;\leq\; g^{'}(x + \Delta x)\Delta x. \tag{8}$$

By (4) and Assumption 1, $K_n > 0$ for every $n = 1, 2, \ldots$.

Consider first part (1) of the proposition. Suppose that $e_{n-1} \geq 0$. By the left inequality of (8), $g_n(\phi_{n-1} + K_n e_{n-1}) \geq g_n(\phi_{n-1}) + g_n^{'}(\phi_{n-1})K_n e_{n-1}$, and hence, and by (3), (1), and (4),

$$\begin{aligned} e_n \;&\leq\; P_s - g_n(\phi_{n-1}) - g_n^{'}(\phi_{n-1})K_n e_{n-1} \\ &=\; P_s - g_n(\phi_{n-1}) - e_{n-1}. \end{aligned} \tag{9}$$

Subtracting and adding $g_{n-1}(\phi_{n-1})$ to the Right-Hand Side (RHS) of (9), and using (3) weith $n-1$, Equation (6) follows.

Next, consider part (2) of the proposition. Suppose that $e_{n-1} \leq 0$. By (1) - (2),

$$e_n \;=\; P_s - P_n \;=\; P_s - g_n(\phi_n) \;=\; P_s - g_n(\phi_{n-1} + K_n e_{n-1}). \tag{10}$$

We next apply Equation (8) with $x = \phi_{n-1} + K_n e_{n-1}$ and $x + \Delta x = \phi_{n-1}$; note that $\Delta x := -K_n e_{n-1} \geq 0$. The left inequality of (8) implies, together with (1), that

$$g_n(\phi_{n-1} + K_n e_{n-1}) \;\leq\; g_n(\phi_{n-1}) + g_n^{'}(\phi_n)K_n e_{n-1}. \tag{11}$$

Consequently, and by (3) and (1),

$$e_n \;\geq\; P_s - g_n(\phi_{n-1}) - g_n^{'}(\phi_n)K_n e_{n-1}. \tag{12}$$

Subtracting and adding $g_{n-1}(\phi_{n-1})$ to (12) we obtain that

$$\begin{aligned} e_n \;&\geq\; P_s - g_{n-1}(\phi_{n-1}) + g_{n-1}(\phi_{n-1}) - g_n(\phi_{n-1}) \\ &\qquad - g_n^{'}(\phi_n)K_n e_{n-1} \\ &=\; \Big(1 - \frac{g_n^{'}(\phi_n)}{g_n^{'}(\phi_{n-1})}\Big)e_{n-1} + g_{n-1}(\phi_{n-1}) - g_n(\phi_{n-1}), \end{aligned} \tag{13}$$

where the last equality follows from (3) and (4). By (1), $\phi_{n-1} \geq \phi_n$ and hence $g_n^{'}(\phi_n) \leq g_n^{'}(\phi_{n-1})$, namely $\frac{g_n^{'}(\phi_n)}{g_n^{'}(\phi_{n-1})} \leq 1$. By Assumption 1 there exists $\alpha \in (0, 1)$, independent of n, such that $\frac{g_n^{'}(\phi_n)}{g_n^{'}(\phi_{n-1})} \geq \alpha$. Defining $\beta = 1 - \alpha$, the left inequality of (7) follows from (13).

The right inequality of (7) is proved in a similar way to (6). By the right inequality of (8), we have that

$$\begin{aligned} e_n = \;&P_s - P_n \;=\; P_s - g_n(\phi_n) \\ &=\; P_s - g_n(\phi_{n-1} + K_n e_{n-1}) \\ &\leq\; P_s - g_n(\phi_{n-1}) - g_n^{'}(\phi_{n-1})K_n e_{n-1} \;= \\ &P_s - g_{n-1}(\phi_{n-1}) + g_{n-1}(\phi_{n-1}) - g_n(\phi_{n-1}) - e_{n-1} \\ &=\; g_{n-1}(\phi_{n-1}) - g_n(\phi_{n-1}), \end{aligned} \tag{14}$$

thereby establishing the right inequality of (7) and completing the proof. ∎

Proposition 2 implies that P_n converges exponentially fast toward a band (tolerance) around the target level P_s, and the width of the band depends on how fast the plant-equation (2) varies. To see this, suppose that there exists $\varepsilon > 0$ such that for every $n = 1, 2, \ldots$, $|g_n(\phi_{n-1}) - g_n(\phi_n)| < \varepsilon$. Then Proposition 2 implies that, for every $n \geq 2$,

$$-\frac{1}{1-\beta}\varepsilon \;\leq\; \liminf_{n\to\infty} e_n \;\leq\; \limsup_{n\to\infty} e_n \;\leq\; \varepsilon. \tag{15}$$

Certainly no perfect tracking can be obtained when the system is time varying, but Equation (15) shows that when

the system varies slowly, namely ε is small, a narrow band can be approached. In particular, when $\varepsilon = 0$, $\lim_{n\to\infty} P_n = P_s$.

Now suppose that the controller's gain K_n is not computed exactly, but rather is estimated by a quantity $\bar{K}_n > 0$. In this case the control equation (1) is modified to the following equation,

$$\phi_n \;=\; \phi_{n-1} + \bar{K}_n e_{n-1}. \tag{16}$$

The following result is an extension of Proposition 2 and its proof is similar and hence omitted.

Proposition 3: Let $\alpha \in (0,1]$ be as in the proof of Proposition 2, namely, for every $\phi_1 \in I$ and $\phi_2 \in I$ such that $\phi_2 \geq \phi_1$, and for every $n = 1, \ldots, \frac{g_n'(\phi_1)}{g_n'(\phi_2)} \geq \alpha$; by Assumption 1 such α exists. For every $n = 1, 2, \ldots,$

1) If $e_{n-1} \geq 0$, then

$$e_n \;\leq\; \Big(1 - \frac{\bar{K}_n}{K_n}\Big)e_{n-1} + \big(g_{n-1}(\phi_{n-1}) - g_n(\phi_{n-1})\big). \tag{17}$$

2) If $e_{n-1} \leq 0$, then

$$\Big(1 - \alpha\frac{\bar{K}_n}{K_n}\Big)e_{n-1} + \big(g_{n-1}(\phi_{n-1}) - g_n(\phi_{n-1})\big) \leq\; e_n \;\leq\; \Big(1 - \frac{\bar{K}_n}{K_n}\Big)e_{n-1} + \big(g_{n-1}(\phi_{n-1}) - g_n(\phi_{n-1})\big). \tag{18}$$

Observe that if $\bar{K}_n = K_n$ then Equations (17) and (18) reduce to (6) and (7) (with $\beta = 1 - \alpha$), respectively.

Suppose that there exists numbers μ and η such that $0 < \mu < \eta < 2$, and suppose that $\mu \leq \frac{\bar{K}_n}{K_n} \leq \eta$ for all $n = 1, 2, \ldots$. Suppose also that there exists $\varepsilon > 0$ such that, for every $n = 2, \ldots, |g_n(\phi_{n-1}) - g_n(\phi_n)| \leq \varepsilon$. Then simple algebra yields the following inequalities,

$$-\frac{1}{\alpha\mu}\varepsilon \;\leq\; \liminf_{n\to\infty} e_n \;\leq\; \limsup_{n\to\infty} e_n \;\leq\; \frac{1}{\mu}\varepsilon. \tag{19}$$

Note that this equation reduces to (15) when the computation of K_n is exact, namely $\bar{K}_n = K_n$. Also, (19) is an extension of one of the convergence results in [2] where the system is linear and $\varepsilon = 0$.

IV. Modeling and Control of a Multicore Power Regulation System

Consider a processor driven by a supply voltage V and operating at a frequency ϕ. The power dissipating at the processor is a function of both voltage and frequency as well as the workload, and denoted by $P(\phi, V, t)$, it has the following form,

$$P(\phi, V, t) = \alpha(t)CV^2\phi + P_L. \tag{20}$$

This equation, derived from basic physical principles, has been established in the literature; see, e.g., [9]. The first term in its RHS, , $\alpha(t)CV^2\phi$, is the dynamic power component resulting from the switching activity, and the second term, P_L, is the static leakage power. The term $\alpha(t)$ is a time-varying workload parameter representing the switching activity of the processor's logic gates, and C is the total processor capacitive load. The leakage power P_L depends on temperature and voltage, but its time-variations for the considered voltage range are much smaller than those of $\alpha(t)$ and hence can be neglected, and P_L is assumed to have a constant value. Equation (20) presents an incentive for selecting low supply voltages, since P depends on V in a quadratic fashion. However, there exists a frequency-dependent bound on how low V can be set. Reducing the supply voltage of CMOS circuits generally increases their propagation delay [14], and this may violate timing constraints requiring all propagation delays to be less than the clock period $\frac{1}{\phi}$. Therefore, manufacturers specify a mapping $V(\phi)$, determined at design time, to guide the selection of voltage levels as a function of frequency. This mapping is nearly affine (linear plus a constant term) and can be adequately approximated via the term

$$V(\phi) \;=\; m\phi + V_0; \tag{21}$$

please see [10], [11]. With this equation we can write P as a function of ϕ and t, and Equation (20) becomes

$$P(\phi, t) \;=\; \alpha(t)CV(\phi)^2\phi + P_L. \tag{22}$$

The control law described in the previous section requires the online calculation of the derivative term $\frac{dP}{d\phi}$, which by (22) has the form $\frac{dP}{d\phi} = \alpha(t)C\Big(V(\phi)^2 + 2V(\phi)\phi\frac{dV}{d\phi}\Big)$. Generally it is impractical to measure or compute $\alpha(t)$, but possible to measure the total power, voltage, and frequency, while the term $\frac{dV}{d\phi}$ can be obtained from via manufacturer or via simulation [10], [11], and P_L can be measured at design time. Now using (22) with (20) yield the following equation,

$$\frac{dP(\phi,t)}{d\phi} = \big(P(\phi,t) - P_L\big)\left(\frac{1}{\phi} + \frac{2}{V(\phi)}\frac{dV(\phi)}{d\phi}\right), \tag{23}$$

which can be used for on-line computation of the derivative term $\frac{dP}{d\phi}$.

In discrete time, Equation (22) yield the following plant equation,

$$P_n \;=\; g_n(\phi_n) \;=\; \alpha_n CV_n^2\phi_n + P_L, \tag{24}$$

where variations in α_n correspond to the time-varying program workloads. Equation (23) yields the following derivative term,

$$g_n'(\phi_{n-1}) \;=\; \big(P_n - P_L\big)\Big(\frac{1}{\phi_{n-1}} + \frac{2m}{V_{n-1}}\Big), \tag{25}$$

and this equation was used in the simulations described in the next section.

V. Simulation Results

This section reports on the results of simulations of an asymmetric multicore processor consisting of two architecturally distinct types of cores - a complex out-of-order core and a simpler two-way superscalar in-order core.

A. Evaluation Platform

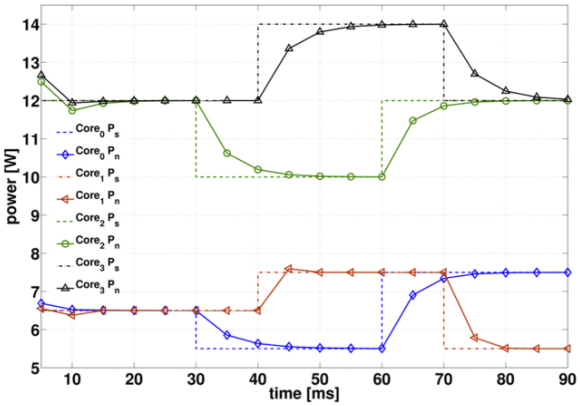
The evaluation platform consists of a cycle-level X86 processor simulator [12] integrated with the McPAT [13] microarchitecture power models. The architectural and physical configurations of the simulated processor are provided in Table I. We simulated the execution of benchmarks programs from the SPEC2006 suite by extracting program traces to drive a 4 core multicore processor interconnected in a 2x2 mesh configuration. The processor is an asymmetric processor with 2 out-of-order cores and 2 in-order cores. Power measurements and controller invocations occur every 5ms. We evaluated the proposed adaptive-gain integral controller and a set of fixed-gain integral controllers with gain values given in K = [25,50,75,100,150,270,385,500]e^6. The initial frequency and supply voltage for each tracking experiment is set to the 3GHz and 0.9V, respectively.

TABLE I
SIMULATED PROCESSOR CONFIGURATION

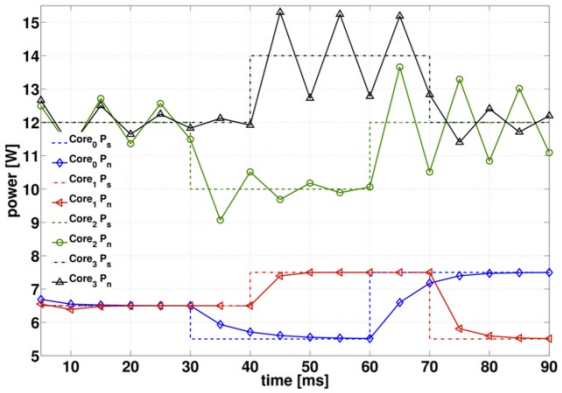
Parameters	Out-of-order Core	In-order Core
Architectural Configuration		
ISA	x86 IA32	
Pipeline Depth	20 stages	16 stages
Fetch/Decode	4 instructions	2 instructions
Execution	6 ports	3 ports
L1 Cache	4-way 32KB	4-way 32KB
L2 Cache	8-way 512KB	8-way 512KB
Physical Configuration		
Clock Frequency	1.85-3.75GHz	
Supply Voltage	0.6-1.0V	
Feature Size	45nm	

TABLE II
POWER TRACKING PHASE FOR ASYMMETRIC PROCESSOR

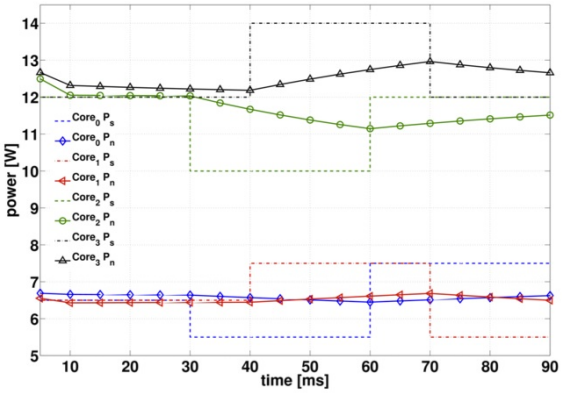
Core	Phase 1	Phase 2	Phase 3
$Core_0$ (in-order)	6.5 W	5.5W	7.5W
$Core_1$ (in-order)	6.5 W	7.5W	5.5W
$Core_2$ (out-of-order)	12 W	10W	12W
$Core_3$ (out-of-order)	12 W	14W	12W



(a) Adaptive gain controller



(b) High fixed gain controller (K = 500e^6)



(c) Low fixed gain controller (K = 25e^6)

Fig. 2. Runtime power tracking results of asymmetric cores.

B. Tracking Analysis

Equations 1-4 were implemented within the simulation model configured as noted in Table I. Figure 2 shows representative runtime power tracking results of the SPEC2006 `milc` benchmark for i) adaptive gain, ii) high fixed gain (K = 500), and iii) low fixed gain (K = 25) controllers. Each core executed the same benchmark and the execution was partitioned into three phases. For each phase the power budget was changed for each core as shown in Table II. The power budgets are shown as dotted lines in the figure. We can observe how well the adaptive gain and static gain controllers track and maintain new power budgets.

The adaptive gain controller tracked the varying reference signals with a time of around 15ms for both in-order and out-of-order cores. The high fixed gain controller is as effective as the adaptive gain controller for the in-order cores but inefficient for the out-of-order cores. The performance difference is due to the microarchitecture heterogeneity between the two cores. Under the same workload, $\frac{dP}{d\phi}$ is greater in the out-of-order case, since it has a higher capacitance C, and can execute more instructions per unit time resulting (larger $\alpha(t)$) compared to the in-order processor. When the power budget is increased, the out-of-order processor will always require a smaller frequency correction Ke_n compared to the in-order case by virtue of it's steeper power vs. frequency relationship. Thus, it expected that a high-enough gain value

may cause significant overshoot in the out-of-order case while being beneficial in the in-order case as shown in Fig. 2. The fact that power budgets can change in unexpected manner as a function of workload demand or even electricity prices (for example based on time of day as is done in data centers), further limits the applicability of fixed gain controllers.

Finally, in principle it may be observed that other static gain values could have produced good tracking properties and possibly better than the examples we have shown. The general observation that a particular controller with a specific gain value can provide good tracking, is of limited value for several reasons. First, from a practical point of view one cannot in general know the applications that will be executed on a platform. Second, the operating system controls where and when threads and processes are scheduled. Extending thread and process schedulers with controller information to constrain scheduling decisions is feasible, but must be weighed against the loss in flexibility and performance that is experienced by limiting the operating system's choices. Third, multicore processors will be executing parallel and not serial applications. Characterization of the power behavior of multithreaded applications on asymmetric processors is still an area of research. Finally, we note that the power and execution time properties of an application can be significantly affected by the input data sets that the applications process. To be practical, extensive off-line analysis of applications to determine the controller gain must cover all possible combinations of core types, applications, and input data sets (the sets that have significant impact on power behavior). We argue that this requirement is limiting and impractical in practice.

By focusing on i) how applications affect fundamental, technology dependent behaviors, namely the frequency-power relationship, and ii) on-line measurement, the adaptive gain controller presented here suffers from none of the preceding drawbacks. Its operation is agnostic to specific applications and core types and thus is a candidate for integration into hardware platforms. However, it does rely on the capability of on-line power measurements. Currently, this is generally not available to user programs at a fine enough granularity in commodity processors. However, there is no significant technical impediment to doing so.

VI. Conclusions

This paper introduced an online controller for processor-power tracking using dynamic voltage-frequency scaling. The proposed control law comprises an integral controller that adjusts its gain in response to changes in the workload to ensure effective regulation and fast settling time. Gain adjustment relies on a novel application-agnostic characterization of the derivative of power that can be cost-effectively done online using power measurements and offline knowledge of the platform's voltage frequency relationship. Tracking property of the proposed algorithm is shown to hold provided that the voltage versus frequency relationship is convex. Simulation results using a cycle-accurate microprocessor demonstrate that the proposed algorithm achieves faster settling times than integral controllers with static gains. The approach holds out promise of applications in the new generation of multicore processor that are asymmetric in the designs of the cores.

References

[1] R.H. Katz, "Tech titans building boom," *IEEE Spectrum*, Vol. 46, no. 2, 2009.

[2] C. Lefurgy, X. Wang, and M. Ware, "Power capping: A prelude to power shifting,"*Cluster Computing,* vol. 11, no. 2, June 2008.

[3] R. Raghavendra, P. Ranganathan, V. Talwar, Z. Wang, and X. Zhu, "No power struggles: coordinated multi-level power management for the data center," *SIGARCH Comput. Architecture News*, Vol. 36, pp. 48-59, 2008.

[4] A.K. Mishra, S. Srikantaiah, M. Kandemir, and C.R. Das, "CPM in CMPs: Coordinated power management in chip-multiprocessors," in *Proc. Intl. Conference on High Performance Computing, Networking, Storage and Analysis*, pp. 1-12, 2010.

[5] R. Kumar et al. Heterogeneous chip multiprocessors. *IEEE Computer*, 38(11), 2005.

[6] T. Morad et al., "Performance, power efficiency and scalability of asymmetric cluster chip multiprocessors," `Compute Architecture Letters`, 2006.

[7] M. Hill and M. Marty, "Amdahls law in the multicore era. IEEE Computer," `IEEE Computer`, 41(7), 2008.

[8] M. Baron, "The Single Chip Cloud Computer,", in *Micrprocessor Report*, April 2010.

[9] M. Floyd, S. Ghiasi, T. Keller, K. Rajamani, J. Rawson, F. Rubio, and M. Ware, "System power management support in the ibm power6 microprocessor," *IBM Journal of Research and Development*, Vol. 51, no. 6, 2007.

[10] T. Burd, T. Pering, A. Stratakos, and R. Brodersen, "A dynamic voltage scaled microprocessor system," in *Proc. Solid-State Circuits Conference*, 2000.

[11] R. McGowen, C.A. Poirier, C. Bostak, J. Ignowski, M. Millican, W.H. Parks, and S. Naffziger, "Power and temperature control on a 90-nm itanium family processor," *IEEE JSSC* Vol. 41, pp. 229-237, 2006.

[12] G.H. Loh, S. Subramaniam, and X. Yuejian, "Zesto: A cycle-level simulator for highly detailed microarchitecture exploration," in *Proceedings IEEE International Symposium on Performance Analysis of Software and Systems* pp. 53-64, 2009.

[13] S. Li, J. Ho Ahn, R. D. Strong, J. B. Brockman, D. M. Tullsen, and N. P. Jouppi, "Mcpat: an integrated power, area, and timing modeling framework for multicore and manycore architectures," in *Proc. IEEE MICRO*, pp. 469-480, 2009.

[14] J.M. Rabaey, *Digital Integrated Circuits: A Design Perspective*, Prentice Hall, 1995.