Robust Optimization of a Chip Multiprocessor's Performance under Power and Thermal Constraints

Mohammad Ghasemazar, Hadi Goudarzi and Massoud Pedram
University of Southern California
Department of Electrical Engineering
Los Angeles, CA 90089 U.S.A.
{*ghasemaz,hgoudarz,pedram*}*@usc.edu*

***Abstract* – Power dissipation and die temperature have become key performance limiters in today's high-performance Chip Multiprocessors (CMPs.) Dynamic power management solutions have been proposed to manage resources in a CMP based on the measured power dissipation, performance, and die temperature of processing cores. In this paper, we develop a robust framework for power and thermal management of heterogeneous CMPs subject to variability and uncertainty in system parameters. More precisely, we first model and formulate the problem of maximizing the task throughput of a heterogeneous CMP (a.k.a., asymmetric multi-core architecture) subject to a total power budget and a per-core temperature limit. Next we develop a solution framework, called Variation-aware Power/Thermal Manager (VPTM), which is a hierarchical dynamic power and thermal management solution targeting heterogeneous CMP architectures. VPTM utilizes dynamic voltage and frequency scaling (DVFS) and core consolidation techniques to control the core power consumptions, which implicitly regulate the core temperatures. An algorithm is proposed for core consolidation and application assignment, and a convex program is formulated and solved to produce *optimal* DVFS settings. Finally, a feedback controller is employed to compensate for variations in key system parameters at runtime. Experimental results show highly promising performance improvements for VPTM compared to the state-of-the-art techniques.**

I. INTRODUCTION

Power dissipation and die temperature have become the main design concerns and key performance limiters in today's high-performance multi-core processors. While design-time approaches exist, the dynamic solution is to utilize a power management unit that takes into account power, performance and temperature of processor cores, and makes the decisions that maximize performance, power efficiency, or both. As CMOS technology scaling continues, intra-die process variations result in higher core-to-core (C2C) power and performance variations. These variations along with device and interconnect aging effects motivate the need to design and deploy robust power management solutions. It is in this context that we intend to tackle the problem of optimizing power efficiency of CMPs under performance, thermal and total power dissipation constraints and subject to different sources of variation.

Versions of the aforesaid problem (e.g., power and temperature constrained performance maximization or performance and temperature constrained power minimization) have been investigated by researchers [1]-[7]. In particular, the authors of [4] present several DVFS (Dynamic Voltage and Frequency Scaling) based techniques to maximize throughput of a homogeneous CMP under a power budget. Some variation-aware algorithms, e.g., linear programming, are presented in [5] for CMP scheduling and power management, to maximize throughput at a given core power budget. However, none of them consider core consolidation, temperature constraint, and leakage dependence on temperature. On the other hand, the authors of [6] study several effective methods for CMP thermal management, such as temperature-tracking frequency scaling, migrating computation to spare hardware units, and a combination of fetch throttling and DVS. The authors of [7] provide an abstract model and convex optimization formulation for speed scaling in multiprocessors under thermal constraints. Closed loop solutions for thermal management of CMPs, such as Model-Predictive Control based solutions [8][9] have been reported as well.

In this work, we consider a heterogeneous CMP performance optimization problem that seeks to maximize the CMP throughput under variations in the system workload and fabrication characteristics of the cores, while the total CMP power consumption is bounded by a given power budget, and the die temperature (estimated by predictive methods or measured by on-chip sensors) is maintained below a critical temperature. We propose a hierarchical power and thermal management for this problem, which utilizes DVFS and core consolidation, and employs a feedback-loop controller. This paper substantially extends our previous work presented in [10] in several major directions: (i) our proposed solution (VPTM) adds the thermal constraint on top of performance constraint; (ii) it formulates and solves the DVFS as a convex optimization problem; (iii) it presents models for temperature, performance and DVFS; and (iv) it solves the core consolidation using a greedy algorithm.

The remainder of this paper is organized as follows. In section II, we present the models we use for CMP. The problem formulation is given in section III, and our proposed solution is explained in section IV. Section V provides our experimental setup and section VI concludes the paper.

II. PRELIMINARIES

In this section we present the models and assumptions that we use in the problem formulation. These models capture the first order effects that are important to the problem, however they are not the most accurate and realistic models and may ignore some second order effects.

This research is supported in part by a grant from the CISE directorate of the National Science Foundation.

A. Throughput and Circuit Delay Model

Task throughput of an application may be defined as the number of application-generated tasks that are serviced per second. It is unwieldy to define throughput of a CMP as the summation of the task throughputs of its running applications because of the potentially drastic differences between the task generation rates and task types of different applications. Also, when dealing with rather long applications, which may execute for a time orders of magnitude longer than power management epoch, *CMP throughput* is typically defined as the summation of instruction throughputs (instructions per second, IPS values) of the active cores. Clearly each core's throughput is a function of its operating frequency. If core *i,* which is running at frequency level *f,* executes application *j* with known characteristics, its throughput can be estimated as,

$$H_i = IPC_i \cdot f_i \tag{1}$$

where IPC_i denotes the instruction-per-cycle (IPC) of core *i*. As a simple model, we *estimate* it as the summation of IPC of all tasks running on this core (assuming that *fast thread switching* method enables efficient use of idle cycles of a task to execute other tasks.) Also, IPC_i can be pre-characterized for each set of tasks when they are consolidated and execute on a core at nominal frequency,

$$IPC_i = \sum_{j\ assigned\ to\ i} IPC_{ij} \tag{2}$$

The IPC of application *j* running on core *i*, IPC_{ij}, depends on the characteristics of the application, its memory access pattern, the core's micro-architecture, and so on [10]. Its value can be obtained by offline profiling or online monitoring of application execution on the target core [13]. Now then, the CMP throughput may be calculated as,

$$H_{CMP} = \sum_i IPC_i \cdot f_i \tag{3}$$

B. Thermal Model

We model the relationship between the die temperature and power dissipation of a core using the thermal model presented in [15]. In our thermal model of the CMP, each node (and the corresponding on-chip temperature predictor/sensor) represents exactly one core. Let $\theta_i(t)$ denote temperature of core *i* at time *t*, and $\boldsymbol{\theta}(t) = [\theta_i(t)]$ ($i = 1, \ldots, N$) denote the vector[1] of temperature readings of all cores at time *t*. Let P_i denote the total power consumption of core *i*, and G_{ij} and G_i represent the thermal conductance between cores *i* and *j* and between core *i* and the ambient, respectively. Using this thermal model, equation (4) calculates the temperature vector at time *t*+1, given the temperature vector, $\boldsymbol{\theta}(t)$, and average power consumption vector, $\boldsymbol{P}(t)$, at time *t*. Note that this calculation is performed periodically, i.e., *t*+1 means one time epoch later than *t*.

$$\boldsymbol{\theta}(t+1) = \boldsymbol{A} \cdot \boldsymbol{\theta}(t) + \boldsymbol{B} \cdot \boldsymbol{P}(t) \tag{4}$$

Note that $\boldsymbol{A}$ and $\boldsymbol{B}$ are matrices containing (empirical) regression coefficients. In this model, the die temperature of a core depends on the die temperatures of other cores, but only the power consumption of the core itself; hence $\boldsymbol{B}$ is a diagonal matrix.

The temperature of any core in the CMP should not go beyond an *emergency temperature*, denoted by θ_{crit}, which is normally provided in the CMP datasheet. Equation (5) is the thermal constraint that is applied to all cores, and equation (6) is its matrix representation in our thermal model.

$$\forall i:\ \theta_i(t+1) \le \theta_{crit} \tag{5}$$

$$\boldsymbol{A} \cdot \boldsymbol{\theta}(t) + \boldsymbol{B} \cdot \boldsymbol{P}(t) \le \theta_{crit} \mathbf{1}_{Nx1} \tag{6}$$

Note that values of elements of matrices **A** and **B** are subject to modeling errors and process-induced variations. In spite of these inaccuracies, we will use matrices **A** and **B** to make a decision about coarse-grain DVFS setting of cores; a closed loop controller will update the DVFS settings in order to avoid any thermal or power violations due to the aforesaid inaccuracies.

C. Voltage and Frequency Relationship

We model the supply voltage of a core, *v*, as a linear function of its frequency, *f*, based on the data extracted from Intel's DVFS technology.

$$v = s \cdot f + v_0 \tag{7}$$

This linear approximation results in a mean square error of less than 5%. Figure 1 illustrates the data extracted from Intel and AMD processors' datasheets [23].

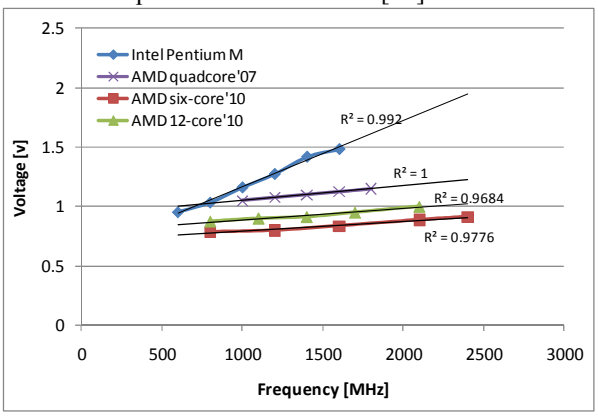

Figure 1. Linear relationship of supply voltage and clock frequency in modern processors.

D. Power Consumption Model

The power consumption of a CMP is the summation of the core power dissipation (the "core power"), plus power dissipation of other shared components on the chip e.g., higher-level caches, memory controller, and other integrated controllers (the "uncore power".) The power manager controls the "core power" by changing the voltage and frequency settings of the cores.

The power dissipation of a core is comprised of dynamic power and leakage power as given below,

$$P_{dynamic}(v,f) = \alpha C_{eff} v^2 f$$
$$P_{leak}(v,\theta) = \eta \theta^2 v \exp\left(\frac{-qV_{th}}{nk\theta}\right) \tag{8}$$

[1] All vector variables are shown in bold face fonts, while scalar variables are in regular fonts.

where θ denotes die temperature and C_{eff}, q, V_{th}, η, n, k, are technology and circuit specific parameters, which can be assumed to be constant, and α is the activity factor which depends on workload. Note that our leakage model, although simple, is quite adequate at this level of optimization.

Due to dependency of leakage power on temperature, there is a positive feedback loop between the die temperature and core power consumption. We neglect the interaction between the supply voltage level and die temperature and assume that they independently affect the leakage power dissipation of a core. Furthermore, since the leakage power consumption is linearly proportional to the core's voltage, v [14], and using a linear approximation of the temperature-dependent component of leakage power (in order to make the computation tractable), we can write a first-order Taylor series approximation of the leakage power as follows:

$$P_{leak}(v,\theta) = k_v \cdot v + k_\theta \cdot \theta \tag{9}$$

According to (7) and (8), dynamic power consumption is super-linearly dependent on the core's clock frequency, f. In addition, the core frequency, f, is linearly related to the core voltage, v. Therefore,

$$P(f,\theta) = d \cdot f^\beta + l \cdot f + k_\theta \cdot \theta \tag{10}$$

where d, l, and k_θ are empirical coefficients for dynamic power consumption, temperature-independent and temperature-dependent components of leakage power dissipation, respectively. Coefficient d, which varies as a function of the workload running on the core, represents the switched capacitance of the core. The β parameter, which has a range between 2 and 3, denotes the exponent of frequency in the dynamic power consumption term (In this report, we assume a β value of 2.5.) Coefficient l is an empirical coefficient relating the temperature-independent component of leakage power dissipation to the core frequency. Values of these coefficients depend on the CMP implementation and fabrication technology parameters. The vector form of the above equation is expressed as,

$$\boldsymbol{P} = \boldsymbol{D} \cdot \boldsymbol{f}^\beta + \boldsymbol{L} \cdot \boldsymbol{f} + \boldsymbol{K_\theta} \cdot \boldsymbol{\theta} \tag{11}$$

in which $\boldsymbol{f}_{\mathrm{Nx1}}$ is the column vector of clock frequencies of cores, and exponentiation in $\boldsymbol{f}^\beta$ is an element-wise operation, which returns a column vector. $\boldsymbol{D}$, $\boldsymbol{L}$ and $\boldsymbol{K_\theta}$ are the diagonal matrices of coefficients d, l, and k_θ of each core.

III. PROBLEM FORMULATION

Consider an N-way, heterogeneous CMP system -such a system is composed of N processing cores [16], which are independent except that the cores share an L2 cache and interface to the main memory. Each core has a separate supply voltage and clock generation module so that the cores can potentially run at different voltage-frequency (v-f) settings, through a supervisory process called per-core DVFS [17].

A power and thermal manager (PTM) performs simultaneous core consolidation and DVFS for cores in the CMP, according to measured core temperatures and power dissipations. More precisely, the PTM seeks to determine the set of ON cores and assign their voltage and frequency levels such that the total CMP throughput is maximized while the following constraints are met: 1) temperatures of cores do not exceed critical temperature (*per-core thermal constraint*), and 2) the total CMP power dissipation is less than the given power budget (*total power constraint*.)

Core consolidation refers to an optimization process by which various running applications are assigned to as few active cores as possible. The hope is that the number of needed cores will be less than the total number of cores on the CMP, and hence, the remaining inactive cores can be power gated and put into a sleep state, and power can be saved while no performance is lost. To employ core consolidation, we define an assignment (mapping) parameter, m_{ij}, which is a pseudo-Boolean variable that represents the assignment of application j to core i,

$$m_{ij} = \begin{cases} 1 & \text{if application } j \text{ is assigned to core } i \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

$$\forall j: \ \sum_i m_{ij} = 1 \tag{13}$$

Note that a core can be turned off only if there are no applications assigned to it, i.e., when $\sum_j m_{ij} = 0$.

Hence, the total CMP throughput can be rewritten as,

$$\begin{aligned} H_{CMP} &= \sum_{\mathrm{i}} \sum_{\mathrm{j}} m_{ij} \cdot IPC_{ij} \cdot f_i \\ &= \boldsymbol{f}^{\boldsymbol{T}} \cdot (\boldsymbol{M} \odot \boldsymbol{IPC}) \cdot \boldsymbol{1}_{J\times 1} \end{aligned} \tag{14}$$

where the symbol $\cdot$ denotes the inner product of two vectors, the symbol $\odot$ denotes element-wise multiplication of two matrices or two vectors of the same dimensionality, $\boldsymbol{f}$ is the vector of frequencies of cores, $\boldsymbol{IPC}$ is the matrix of IPC of applications if executed on any core, and $\boldsymbol{M}$ is the matrix of application to core assignment variables, m_{ij}.

Now then, the problem statement can be written as a mixed-integer program, as illustrated in (15), which applies to both homogeneous and heterogeneous CMP architectures. In this formulation, the objective function is to maximize the throughput by using the model of (14), whereas the constraints are the thermal emergency constraint given by (6), the total CMP power budget given by (11), and a constraint on the maximum and minimum limits on frequency.

$$\begin{cases} Maximize\ H = \boldsymbol{f}^{\boldsymbol{T}} \cdot (\boldsymbol{M} \odot \boldsymbol{IPC}) \cdot \boldsymbol{1}_{J\times 1} \\ \text{subject to:} \\ \quad \boldsymbol{A} \cdot \boldsymbol{\theta} + \boldsymbol{B} \cdot \boldsymbol{P} \le \theta_{\boldsymbol{crit}} \boldsymbol{I} \\ \quad \boldsymbol{1} \cdot \boldsymbol{P} \le P_{budget} \\ \quad \boldsymbol{f_{min}} \le \boldsymbol{f} \le \boldsymbol{f_{max}} \\ \quad \boldsymbol{1}_{1\times N} \cdot \boldsymbol{M} = \boldsymbol{1}_{1\times J} \\ \quad \boldsymbol{P} = \boldsymbol{D} \cdot \boldsymbol{f}^\beta + \boldsymbol{L} \cdot \boldsymbol{f} + \boldsymbol{K_\theta} \cdot \boldsymbol{\theta} \end{cases} \tag{15}$$

The above problem formulation is a mixed integer program, which is NP-hard. This is because the multi-processor job assignment problem, which is known to be NP-hard [18], can be reduced to (15). Next, we will present a simpler version by relaxing some constraints, to be able to solve it efficiently.

IV. PROPOSED SOLUTION

Our proposed Variation-aware Power Thermal Manager (VPTM) consists of four modules: a *tier-one manager* (T1-PTM), a *tier-two manager* (T2-PTM), a *proportional-integral* (PI) *feedback controller* [19], and a *Workload Analyzer Unit* (WAU.) Figure 2 illustrates the architecture of the proposed Variation-aware Power and Thermal Manager, *VPTM*.

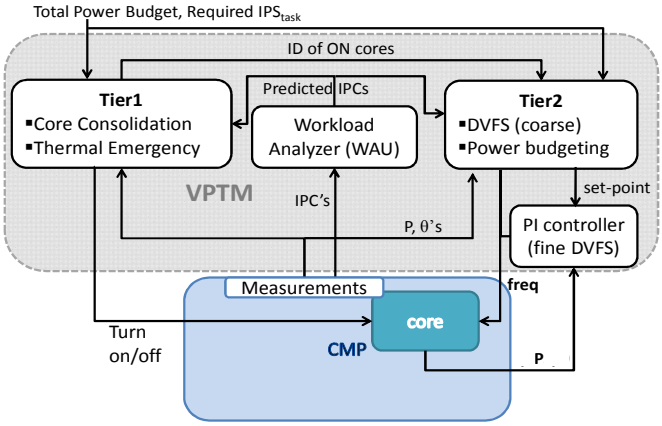

Figure 2. Block diagram of VPTM.

T1-PTM performs core consolidation and identifies the cores to be turned off in order to increase power efficiency of the CMP and resolves the thermal emergencies (when die-temperature reaches the critical temperature.) The T2-PTM uses this information to *sub-optimally* decide on the frequency of cores for the next epoch. It also calculates the set points for the (hardware-based) controller of core for the next epoch. The PI controller *fine-tunes* the core DVFS settings based on actual measurements at runtime. It makes VPTM robust to variations of workload, as well as PVT.

WAU analyzes the workload and predicts its characteristic, i.e., IPC of applications, for the next decision epoch. WAU continuously monitors the actual IPC of applications (using core's performance counters [13]), and applies a moving average calculation to update the IPC values for the next epoch, which reduces the estimation error caused by workload variation. This predicted IPC data is passed onto T1-PTM and T2-PTM next. We use a moving average predictor in this work; however, utilization of more accurate prediction algorithms increases the quality of predicted values at the cost of increased complexity.

A. Tier-One Manager – Core Consolidation

In tier-one of the VPTM, T1-PTM, we adopt a heuristic to perform core consolidation decisions and avoid thermal emergencies (when a core temperature rises above θ_{crit}) at the beginning of each *decision epoch* of duration T_1. As will be detailed later, Tier-two of VPTM finds the optimal DVFS configuration of active cores such that all constraints, including the thermal constraint that no core temperature rises above θ_{crit}, are satisfied at the beginning of a *timing window* of size $T_2 = T_1/k$ where k is a small natural number, say 2 or 3. Now then, under this hierarchical control architecture, it is possible that the tier-two DVFS is not capable of keeping the core temperature below the critical temperature value, θ_{crit}. In this case, the core temperature approaches the critical temperature value, θ_{crit} . This is due to the fact the T2-PTM cannot turn off any core since that decision is reserved for the T1-PTM, which runs at each decision epoch. To be safe, we thus impose the constraint that no core temperature exceeds a *threshold temperature*, denoted by θ_{th}. Note that $\theta_{th} < \theta_{crit}$ .This imposes limits on T_1 and T_2 since we have to ensure that, even in the worst case, the temperature of a core cannot rise from θ_{th} to θ_{crit} in time T_1 if we set its voltage and frequency levels to the minimum allowed after time T_2. Note that a core whose temperature is above θ_{th} and rising towards θ_{crit}, will be turned off for the next few epochs (and the applications running on it will be migrated to other cores), until it cools down below a second temperature value, $\theta_{cool} < \theta_{th}$, and only then, it may be turned back on.

The proposed T1-PTM is a greedy (steepest-ascent) hill-climbing algorithm that seeks to reach a local optimum solution by gradually moving towards the maximum point in a solution space neighborhood. A neighbor of the current solution is defined as one of the following three cases: (i) a solution with the same number of active cores, (ii) a solution with one more active (also called ON) core, or (iii) a solution with one fewer active core. Now then, the proposed algorithm explores the neighbors (in terms of the number of ON cores) of the current system configuration, and if it finds a better solution (yielding higher CMP throughput while meeting the power budget), it chooses and enforces that solution for the next decision epoch. T1-PTM relies on the quality of estimates of the CMP throughput, power dissipation, and die temperature at the end of current epoch, using the predicted data provided by WAU.

The key idea for core consolidation is to group low IPC applications that may be running on two or more cores into one core whenever possible, and to turn off the other cores (or set them to some lower power state) resulting in noticeable power saving. Assuming fast thread switching support (similar to the Sparc family and Niagara architectures [20]), the performance overhead of core consolidation is negligible. At the same time, T1-PTM may have to migrate applications from some active core to another active core or even to turn on a new core in order to maximize IPS. Note that in case of memory-bound low IPC applications, conflicting cache misses may decrease the advantages of core consolidation, if the cache size is small.

To perform consolidation, T1-PTM calculates the IPC of each core as a weighted summation of IPC's of all of the applications that are running on the core. Next it sorts the active cores in ascending order of their IPC values and creates an active core queue. It then examines the first pair of cores in this queue (i.e., those with the lowest IPC values.) It checks to determine if the tasks running on these two cores can be consolidated into one of the cores without violating power or thermal constraints. If so, this consolidation is performed. If not the next pair of cores from the active core queue is considered as a consolidation candidate. The process continues until a pair of cores is found or the queue is completely processed.

Similarly, if a core is running more than one application and it is at the maximum core frequency or a thermal emergency can be created, the core is a candidate for migrating one or more of its applications to some other core (we call this process "core de-consolidation".) Note that migration of applications between cores has latency and energy overhead, which is taken into account when considering consolidation and de-consolidation actions.

Note that the described consolidation is possible only for cores with similar architecture (but different performance and power), and tasks should use similar ISA.

B. Tier-Two Manager – Coarse-Grain DVFS

T2-PTM solves a simplified version of the mixed-integer problem (15) by eliminating the pseudo-Boolean assignment variables (since they have already been determined by T1-PTM.) Hence T2-PTM solves the nonlinear program of (16), which maximizes the total CMP throughput while satisfying the aforesaid constraints. The problem of (16) is a convex optimization problem, and an optimal solution can be found in polynomial time. The problem is indeed a modified version of the convex problem presented and solved online in [22], where the objective function was the summation of cores' frequencies. In contrast, our objective function is the actual CMP throughput, which is a weighted summation of the core frequencies. We will use the solution method presented in [22] to efficiently solve (16). Note that to convert the problem to a convex one, the last constraint of (16) has been replaced with an inequality, however, the optimum solution will be same as if it is an equality (see [22] for details.) Also note that in this formulation we use θ_{th} instead of θ_{crit}.

$$\begin{cases} Maximize\ H = \boldsymbol{f} \cdot \boldsymbol{IPCA} \\ \text{subject to:} \\ \quad \boldsymbol{A} \cdot \boldsymbol{\theta} + \boldsymbol{B} \cdot \boldsymbol{P} \le \theta_{th}\boldsymbol{1} \\ \quad \boldsymbol{1} \cdot \boldsymbol{P} \le P_{budget} \\ \quad \boldsymbol{f_{min}} \le \boldsymbol{f} \le \boldsymbol{f_{max}} \\ \boldsymbol{P} \ge \boldsymbol{D} \cdot \boldsymbol{f}^{\beta} + \boldsymbol{L} \cdot \boldsymbol{f} + \boldsymbol{K_{\theta}} \cdot \boldsymbol{\theta} \end{cases} \quad (16)$$

Note the implicit assumption of running an application at no more than a single core at any given time. Note that if the temperature of a core has exceeded the critical temperature and reducing its frequency level to the minimum does not stop the rise towards the critical temperature, then T1-PTM turns the core off at the beginning of the next decision epoch.

C. Tier-Three Manager – Closed-loop Controller

Despite the global optimality of the solution to convex optimization problem of (16), it ignores the variation and uncertainty in the characteristics of cores and behavior of applications, such as regression coefficients of power consumption and IPC of applications. As a result, a direct solution of in (16) may suffer from overestimating or underestimating temperature, power, or throughput.

Thus, to be suitable as a *robust online power management*, VPTM utilizes a PI (Proportional-Integral) controller [19] for each core to dynamically adjust the frequency of the core so as to maintain its per-core power budgets close to their desired values, in spite of potential changes in the application behavior. This requires a break-up of the total CMP power budget to target power budgets for all active cores, a step which we do by setting the per-core power targets at the level required by core's calculated frequency and temperature in the optimal solution to (16).

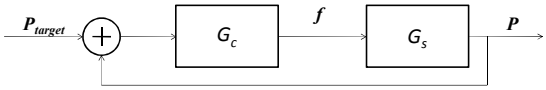

Figure 3. PI controller of VPTM.

As illustrated in Figure 3, each core (*Gs*) has a controller (G_c) that set its frequency. Based on the solution foundby T2-PTM, the target power budget of each core is set and to maintain its power dissipation at the desired level. The PI controller continuously measures the actual core power dissipation, and if required, changes its DVFS setting to match the set point, determined by T2-PTM. Details of designing this controller (i.e. setting its paramteres) follows the conventional PI controller design approach to guarantee its stability and response quality.

V. EXPERIMENTAL RESULTS

For our experiments, we setup a tool chain, which is an in-house MATLAB-based CMP simulator integrated with PTscalar, a cycle-accurate microarchitecture level power, performance, and thermal simulator (it uses a temperature-dependent leakage model) [21]. Multiple instances of PTscalar simulate execution of tasks on cores, and calculate the power and temperature of cores at each time epoch, then these values are reported to our PTM unit (in MATLAB) which decides core consolidation and task migration moves and adjusts DVFS settings of cores. We simulate a heterogeneous quad-core CMP in which the cores are of two types that vary in architecture and operating frequency. Cores 1 and 2 are faster (they run at 3.2GHz and have a larger issue and commit queue) while cores 3 and 4 are slower (they run at 2.6GHz and have smaller queues) -in the problem formulation, each individual core can be of any arbitrary type. Both core types are similar to Alpha 21264 architecture. The architecture of cores in our experiments is similar to Alpha 21264 architecture, with some changes in the configuration and parameters, as listed in Table 1. The ambient temperature is set to 25°C, and the critical temperature is set to 100°C.

Table 1. Configurations of the cores in CMP system

Pipeline	Out-of-order
Fetch-Issue-Commit	4-4-4/4-2-2
Load/Store queue	32/32
L1 instruction/data cache	16KB, 2-way/8KB, 2-way/LRU
L2 unified cache	4MB, 8-way, 64B line
Technology node/Vdd	32nm, 1.1V/1V
Max frequency	3.2GHz/2.6GHz/2.3GHz

We first used PTscalar to extract the thermal and power model parameters, i.e. $\boldsymbol{D}$ (per task), $\boldsymbol{L}$, $\boldsymbol{K_{\theta}}$, $\boldsymbol{A}$ and $\boldsymbol{B}$ matrices. Then, the effect of process variation was estimated by applying up to 5% random deviation to these parameters that are being used in the PTM solver.

For workload, we use bundles of four different benchmarks selected from SPEC2000 benchmark suite (as mentioned earlier, we do not consider inter-task

communication in this work.) The task mix is assigned to CMP and run virtually forever. Execution of each task on any core type is pre-characterized in terms of its average IPC, *D*, and *L* values. Note that these values are used as uncertain data and VPTM uses a moving average (MA) predictor (of length three) and a feedback loop to manage uncertainties. Table 2 illustrates a sample assignment of tasks to cores, and resulting average IPC of tasks on corresponding cores.

Table 2. Assignment of benchmarks in test1

Core	**1**	**2**	**3**	**4**
Benchmark	twolf	mcf	equake	bzip
Avg. IPC	1.205	2.12	1.7	0.90

Figure 4 demonstrates the performance of VPTM algorithm for the benchmark set and its given assignment in Table 2. Our baseline is a greedy algorithm called PushHiPullLo (PHPL) which is similar to the greedy algorithm presented in [3]. PHPL maximizes CMP throughput under a total power budget by consecutively reducing the frequency of the core with lowest IPC, until the power budget is met. Limiting the maximum frequency of cores enforces the thermal constraint. In Figure 4, plot (a) demonstrates simulated CMP throughput and power. In this experiment, we have applied a sequence of {110W, 80W, 100W, 80W} for total power budget. This sequence not only shows the behaviour of VPTM for two high and low power budgets, but also demonstrates the transition between these two states and the settling time. Plots (b) and (c) illustrate trace of frequency and temperature (θ_{crit}=100) of each core, respectively. As it can be seen, VPTM follows the power budget very closely, which is because of the PI-controller, that adaptively updates DVFS to maintain target core powers. Another observation is that in VPTM, core 3 is executing a task with a high IPC (but not the highest) while its power is the most proportional, and hence the maximum power budget is allocated to it, and its frequency is mostly at its maximum.

Figure 5 demonstrates performance of PHPL. For purpose of comparison, we disabled core consolidation capability of

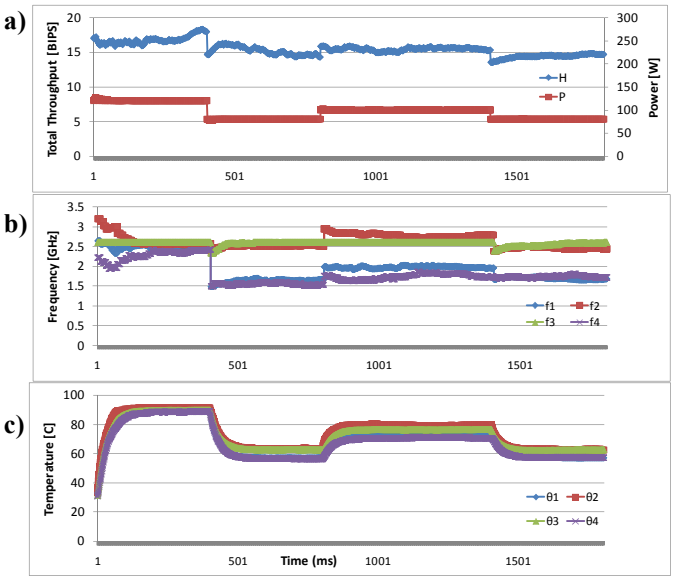

Figure 4. Performance of VPTM algorithm.

tier-one of VPTM, since the comparison baseline considered here does not perform core consolidation (to the best of our knowledge, there is no algorithm in the literature that solves the same problem as VPTM, by a combination of DVFS and core consolidation.) In Figure 5, plot (a) demonstrates simulated "measured" CMP throughput and power consumption. In this experiment, we have applied a sequence of {110W, 80W, 100W, 80W} for total power budget. Comparison of this plot with the one in Figure 4 shows that the average total IPS of VPTM is higher than PHPL for similar power budgets (which are 15.5 and 13.2 BIPS, respectively.) Plots (b) and (c) of these figures illustrate trace of frequency and temperature (θ_{crit}=100) of each core, respectively. Also, as it can be seen, VPTM follows the power budget very closely, which is because of the PI-controller, that adaptively updates DVFS to maintain target core powers. In PHPL in contrast, core 2 has the highest IPC and mostly runs at its maximum frequency.

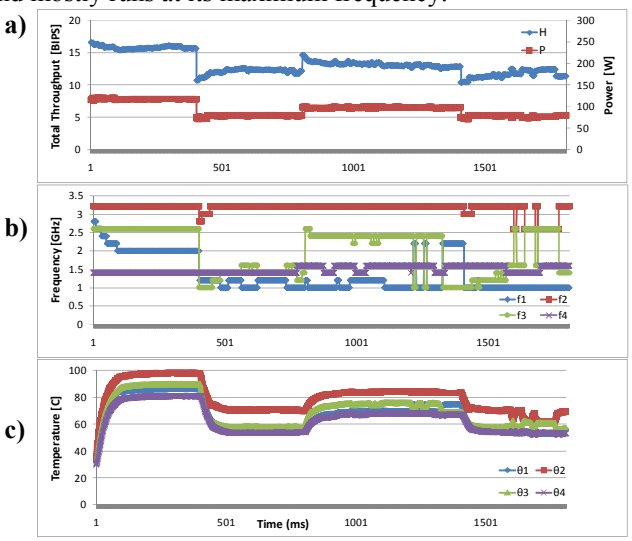

Figure 5. Performance of PushHiPullLo algorithm.

Figure 6 compares average performance of VPTM (with core consolidation enabled) to that of PHPL for five different mixes of benchmark selection, under three power budget conditions. The average throughput of VPTM is approximately 21.4% higher than PHPL. An average of 13% is gained by combination of precise solution of DVFS and utilization of PI controllers, and the rest is due to core consolidation.

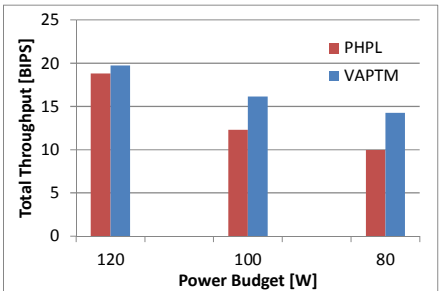

Figure 6. Total IPS under power budget –VPTM vs PHPL.

We also studied performance of VPTM on eight-core CMPs consisting of two 3.2GHz cores, two 2.6GHz and four

2.3GHz cores, which all have similar architectural configuration as in the previous experiments. We compared performance of VPTM in eight-core CMPs to PHPL, as depicted in Figure 7. and Table 3.

Table 3. Total throughput of 8core CMPs at different power budgets

Pbudget [W]	150	125	100	75	50	30
PHPL	34.01	31.29	28.43	23.82	17.94	15.55
VPTM	34.26	33.25	30.83	27.56	22.90	17.83
% Improvement	0.72	6.28	8.47	15.69	27.65	14.65

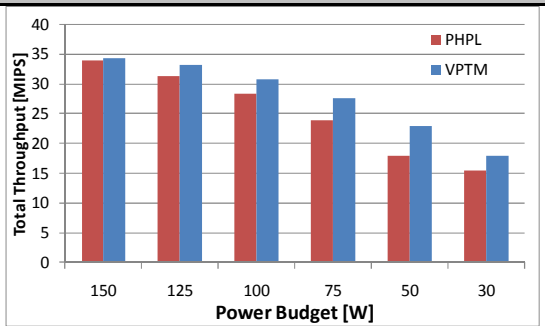

Figure 7. Comparison of VPTM and PHPL in 8-core CMPs.

As it can be observed, at very high power budgets, the total throughput of VPTM and PHPL are similar, while at mid-range power budgets, they substantially differ. The reason is that at high power budgets, all cores are running at their maximum frequency; hence there is no room for optimization. However, at mid values of power budget, the VPTM optimally sets the DVFS setting and achieves up to 18% better throughput than PHPL. The runtime of VPTM is determined by runtime complexity of tier-1 (consolidation) plus runtime of solving (16). The complexity of consolidation step is $O(N.logN)$ where N denotes the number of cores, and its decision epoch is in the order of tens of milliseconds. Runtime of solving (16), which is invoked in the order of operating system's 10ms time slice, is about 50-100μs; i.e., less than 1% performance overhead. Finally, PI-controller performs few simple arithmetic calculations every hundreds of microseconds. This makes VPTM runtime acceptable as an online PTM.

VI. CONCLUSION AND FUTURE WORK

We presented a mathematical formulation and solution to the problem of power and thermal management in heterogeneous CMPs by proposing a hierarchical Variation-aware Power and Thermal Manager (VPTM). VPTM maximizes throughput of a CMP operating under variations/uncertainties by means of DVFS and core consolidation, subject to a given total power budget, and a constraint on die temperature. PI controller was employed to compensate for variations in key system parameters at runtime. Experimental results of VPTM show up to about 20% performance improvements, with no impact on the maximum temperature, for a given power budget.

As part of our future work, we intend to extend the VPTM framework where the system level performance objective is the average response time per task rather than overall instruction throughput (IPS) of each task. This extension is an important one for many high-end servers and hosting datacenters where the end user cares about the latency. Moreover, we will improve on some of the models and assumption that we have used and consider more realistic models, including thermal model, and a more precise model for estimation of IPC of consolidated tasks. In addition, we will apply and extend the VPTM approach to a virtualized multi-core server system.

REFERENCES

[1] J. Donald and M. Martonosi, "Techniques for Multicore Thermal Management: Classification and New Exploration," *SIGARCH Computer Architecture News,* 2006.

[2] J. Sharkey, A. Buyuktosunoglu, P. Bose, "Evaluating Design Tradeoffs in On-Chip Power Management for CMPs," *ISPLED*, 2007.

[3] S. Herbert, D. Marculescu, "Analysis of dynamic voltage/frequency scaling in chip-multiprocessors," *ISLPED*, 2007.

[4] C. Isci, *et al.* "An Analysis of Efficient Multi-Core Global Power Management Policies: Maximizing Performance for a Given Power Budget," *Proc. IEEE/ACM int'l Symp. on Microarchitecture,* 2006.

[5] R. Teodorescu, J. Torrellas, "Variation-Aware Application Scheduling and Power Management for Chip Multiprocessors," *ISCA*, 2008.

[6] K. Skadron, *et al.* "Temperature-Aware Microarchitecture: Modeling and Implementation," ACM *Transactions on Architecture and Code Optimization*, Vol. 1, No. 1, March 2004, Pages 94–125.

[7] A. Mutapcic, S. Boyd, S. Murali, D. Atienza, G. De Micheli, and R. Gupta "Processor Speed Control with Thermal Constraints," *IEEE Trans. on Circuits and Systems—I*, Vol. 56, NO. 9, 2009.

[8] Y. Wang, K. Ma, and X. Wang, "Temperature-Constrained Power Control for Chip Multiprocessors with Online Model Estimation," *ISCA*, 2009.

[9] A. Bartolini, M. Cacciari, A. Tilli, L. Benini, "A Distributed and Self-Calibrating Model-Predictive Controller for Energy and Thermal management of High Perfromance Multicores," *DATE*, 2011.

[10] M. Ghasemazar, E. Pakbaznia, and M. Pedram, "Minimizing the power consumption of a chip multiprocessor under an average throughput constraint," *Proc. of the 11th Int'l Symposium on Quality of Electronic Design*, Mar. 2010, pp. 362-371.

[11] M. Aater Suleman, O. Mutlu, M.K. Qureshi, and Y.N. Patt, "Accelerating Critical Section Execution with Asymmetric Multicore Architectures," *IEEE Micro*, Vol.30, No.1, 2010.

[12] E. Humenay, D. Tarjan, and K. Skadron, "Impact of Process Variations on Multicore Performance Symmetry," *DATE,* 2007.

[13] Intel® 64 and IA-32 Architectures Software Developer's Manual, 2009, http://www.intel.com/products/processor/manuals/ [online]

[14] M. Weiser, *et al.*"Scheduling for reduced CPU energy," *Proc.USENIX Symp. on Operating Systems Design and Implementation*, 1994.

[15] Y. Han, I. Koren, C. M. Krishna, "TILTS: A fast architectural-level transient thermal simulation method," *Journal of Low Power Electronics*, 3(1), 2007.

[16] R.Kumar, D. M. Tullsen, N. P. Jouppi, P. Ranganathan, "Heterogeneous Chip Multiprocessors," *IEEE Computer*, 38, 2005.

[17] W. Kim, M. Gupta, G. Y. Wei, D. Brook, "System level analysis of fast, per-core DVFS using on-chip switching regulators,"*HPCA,* 2008.

[18] H. Aydin, Q. Yang, "Energy-Aware Partitioning for Multiprocessor Real-Time Systems," *Int'l Symp. Parallel &Distributed Processing*, 2003.

[19] R. C. Dorf, R. Bishop, *Modern Control Systems*, Prentice Hall, 2008.

[20] P. Kongetira, K. Aingaran, and K. Olukotun, "Niagara: A 32-Way Multithreaded SPARC Processor," *IEEE MICRO Magazine*, 2005.

[21] W. Liao, L. *He,* K. M. Lepak, "Temperature and Supply Voltage Aware Performance and Power Modeling at Microarchit*ecture Level*," *IEEE Trans. Computer-Aided Design*, 24:1042–1053, 2005.

[22] S. Murali, *et al.*,"Temperature-Aware Processor Frequency Assignment for MPSoCs Using Convex Optimization," IEEE/ACM *Hardware/software Codesign and System Synthesis*, 2007

[23] Advanced Micro Devices, Family 10h AMD Opteron Processor Product Data Sheet, Revision: 3.04, [online] http://support.amd.com/us/Processor_TechDocs/40036.pdf