Run-Time Thermal Management of Three-Dimensional Chip-Multiprocessors

Changyun Zhu[1] Zhenyu (Peter) Gu[2] Li Shang[3] Robert P. Dick[4] Russ Joseph[4]

[1]ECE Department Queen's University
[2]Synplicity
[3]ECE Department University of Colorado at Boulder
[4]EECS Department Northwestern University

***Abstract*—Three-dimensional (3D) integration has the potential to improve the communication latency and integration density of chip-level multiprocessors (CMPs). However, the stacked high power density layers of 3D CMPs increase the importance and difficulty of thermal management. In this paper, we investigate the 3D CMP run-time thermal management problem and describe efficient management techniques. This work makes the following main contributions: (1) it identifies and describes the critical concepts required for optimal thermal management, namely the methods by which heterogeneity in both workload power characteristics and processor core thermal characteristics should be exploited and (2) it proposes an efficient, proactive, continuously-engaged hardware and operating system thermal management technique governed by optimal thermal management polices. The proposed technique is evaluated using multi-programmed and multithreaded benchmarks in an integrated power, performance, and temperature full-system simulation environment. We find that proactive power–thermal budgeting allows a 30% improvement in instruction throughput compared to a state-of-the-art proactive thermal management approach that bases decisions only upon local information. The software components of the proposed thermal management technique have been implemented in the Linux 2.6.8 kernel. This source code will be publicly released. The analysis and technique developed in this work provide a general solution for future 3D and 2D CMPs.**

I. Introduction

Continued increases in integration density, and achieving higher application performance without corresponding increases in processor frequency, are now primary goals for microprocessor designers. As a result, microprocessor design is rapidly moving towards highly-scalable chip-multiprocessor (CMP) architectures. Today's mainstream microprocessors are multi-core [1], [2], [3], [4], [5], [6]. The trend for future CMPs is to increase the number of on-chip cores: 80-core prototypes have recently been demonstrated by Intel [7].

Performance scalability is a major challenge in CMP design. Using the mainstream two-dimensional (2D) planar CMOS fabrication process, on-chip interconnect shows poor scalability in both performance and power consumption [8]. Three-dimensional (3D) integration has the potential to overcome the limitations of 2D technology [9], [10], [11], [12]. By stacking multiple device layers connected with inter-die vias, 3D integration increases logic integration density significantly and reduces on-chip wire length, especially for global and semi-global wires. This has motivated computer architects to evaluate 3D technology for CMP architecture design [10], [13], [14], [15]. However, none of this work describes a thermal management solution appropriate for 3D CMPs, nor has prior work described the requirements for optimal power–thermal budgeting in 2D or 3D CMPs.

Thermal issues are a large and growing concern for CMPs [16], [17], [18], [19]. Increasing chip power consumption and temperature affect circuit reliability (via negative bias temperature instability, electromigration, time-dependent dielectric breakdown, thermal cycling, etc.), power and energy consumption (via increased leakage power), and system cost (via increased cooling and packaging cost). Robustness to temperature-dependent timing errors poses a particularly-interesting problem for thermal management because the most efficient power control techniques frequently have high latencies. This results in a trade off between power–performance efficiency and reliability, and often implies the use of multiple power control techniques within a thermal management infrastructure.

The use of 3D integration magnifies power dissipation problems [10], [20], [21], [22]. Chip cross-sectional power density increases linearly with the number of vertically-stacked active circuit layers. 3D integration holds promise but without solutions to the thermal problems it brings, 3D CMPs will be impractical.

Run-time thermal management techniques, such as dynamic voltage and frequency scaling, clock throttling, execution unit toggling, and workload migration, have been proposed for high-performance microprocessors [18], [19], [23], [24], [16], [17]. Using these techniques, cooling solutions and packages need not be designed for worst-case power consumption scenarios. Cooling cost can thereby be significantly reduced. Past work, however, cannot effectively optimize the performance–temperature tradeoff in 3D CMPs for the following reasons.

First, the thermal management techniques deployed in current microprocesasors and operating systems are primarily used to handle rare, worst-case power consumption events and eliminate thermal emergencies. Although they can potentially introduce significant performance overhead, they are rarely invoked. In contrast, the higher power densities of future 3D (and some 2D) CMPs will frequently require operation at or near thermal limits. Already, processors contain reactive techniques to permit the use of reduced-cost packaging and cooling

This work was supported in part by the NSERC Discovery Grant #388694-01; in part by the National Science Foundation under awards CNS-0347941, CNS-0702761, and CNS-0720691; and in part by the Semiconductor Research Corporation under award 2007-HJ-1593.

configurations that are not capable of handling maximum power dissipation. Today's laptops frequently invoke thermal management mechanisms that drastically reduce performance, even under normal operating conditions [25]. Power should be viewed as a limited resource and processor cores should spend carefully-budgeted amounts. Thermal management should be used to proactively, continuously optimize CMP performance and temperature, instead of merely reacting to emergencies.

Second, 3D CMPs have heterogeneous power and thermal characteristics. On-chip processor cores have different cooling efficiencies. For instance, cores in the layers closer to the heatsink have higher cooling efficiencies than those farther from the heatsink. Processor cores farther from the heatsink will have higher temperatures than their neighbors nearer the heatsink, even when their power consumptions are lower. Inter-core thermal correlation is heterogeneous. The thermal correlation between vertically-aligned processor cores is stronger than that between processor cores within the same layer. The power and thermal heterogeneity of 3D CMP poses unique challenges for run-time thermal management. Achieving optimal 3D CMP performance under a temperature constraint requires careful system-wide control of each processor core's performance and power consumption. Local control, alone, is insufficient.

In this article, we develop the analytical framework necessary to determine the thermal impact of every core in a 3D CMP upon every other core. This framework yields guidelines for near-optimal thermal management. The guidelines are embodied in a proactive global power–thermal budgeting algorithm, performance counter-based workload monitor, and distributed thermal control techniques, which we have implemented in version 2.8.6 of the Linux kernel; this code will be publicly released. The resulting 3D CMP thermal management solution, which we call ThermOS, is evaluated using detailed full-system simulation with M5 [26]. We have integrated power modeling and thermal analysis tools within the simulator, allowing unified architectural/power/thermal simulation of arbitrary single-threaded and multi-threaded applications and the Linux operating system (OS). Our results for a wide range of multiprogrammed and multithreaded applications indicate that, given a peak temperature constraint, ThermOS improves CMP throughput by an average of 30% when compared to state-of-the-art proactive distributed thermal management. This improvement is primarily due to the power–thermal budgeting guidelines used by ThermOS.

II. Heat flow in 3D CMPs

This section uses examples to explain the special thermal characteristics of 3D CMPs and develop a mathematical model that will be used to derive the thermal management policies described in Section III and validated in Section IV.

A. Introduction to Thermal Modeling

Heat conduction within CMP chip and package can be modeled using Fourier heat flow analysis, which has been the standard method used by industry and academia for circuit-level and architecture-level IC chip–package thermal analysis during the past few decades [27], [28], [19], [29].

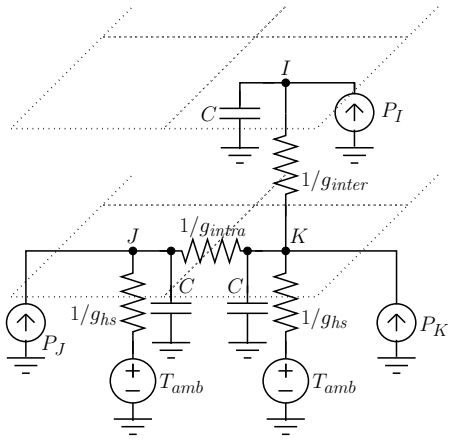

Fig. 1. Inter-layer and intra-layer thermal heterogeneity and dominance in 3D CMPs.

This method is analogous to Georg Simon Ohm's method[1] of modeling electrical current. Using Fourier heat flow analysis, heat flow is analogous to electrical current and temperature is analogous to voltage. The CMP is virtually partitioned into numerous discrete blocks, as shown in Figure 1 (Figure 2 gives a higher-level overview of the 3D CMP structure). The thermal conductance of each block is a linear function of the conductivity of its material and its cross-sectional area divided by length; it is analogous to electrical conductance. Blocks also have heat capacities that are analogous to electrical capacitance. Therefore, an instantaneous change in heat generation results in a gradual change in temperature. As a result, the thermal profile of a CMP is essentially its power profile after applying a complicated RC filter. For a thermal model to be accurate, each block must be so small that the temperature within it is uniform. A fine-grained, and thus more accurate, model was used to validate ThermOS. However, for the sake of explanation, this section will describe the coarse-grained model shown in Figure 1, in which each core is represented with a single thermal model element.

In 3D CMPs fabricated from multiple stacked wafers, the thermal environment varies from layer to layer. Moreover, the intra-layer and inter-layer thermal relationships among CMP cores are heterogeneous. The rest of this section explains the impact of this heterogeneity on heat flow and builds the theoretical foundations for developing near-optimal 3D CMP thermal management policies.

Homogeneous Intra-Layer Characteristics: Figure 1 illustrates a simplified heat conduction model for a pair of adjacent CMP cores on the same layer (J and K) and a pair of adjacent CMP cores on different layers (I and K) of a 3D CMP. As shown in this figure, since the heat dissipation paths of Cores I and K are nearly identical, the thermal conductances of these two cores to the ambient are nearly equal. In other words, processor cores within the same layer have similar cooling efficiencies.

Heterogeneous Inter-Layer Characteristics: In contrast to cores on the same layer, Cores I and K have different conductances to the ambient for Core K (g_{hs}) and for Core I

[1]In fact, Ohm borrowed this model from Fourier and it was initially proposed to model heat flow.

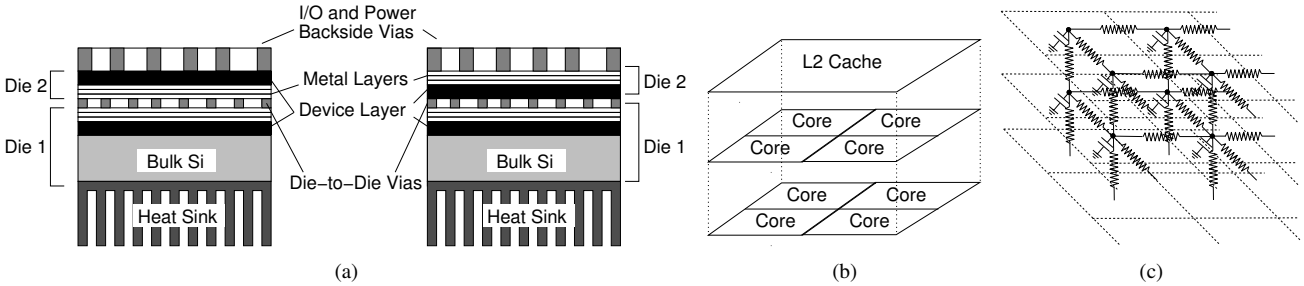

Fig. 2. (a) Comparison of face-to-face (left) and face-to-back (right) configurations for two stacked dies, (b) 3D three stacked die floorplan used in this work, and (c) 3D CMP chip-package thermal modeling.

$(1/(1/g_{hs} + 1/g_{inter}))$. In addition, the steady-state temperature of Core I is always higher than that of Core K, even if Core I has a lower power consumption. The following equations formalize this effect, which we refer to as *thermal dominance*. Neglecting the limited intra-layer heat flow,

$$T_K = T_{amb} + (P_K + P_I)/g_{hs} \quad (1)$$

$$\begin{aligned} T_I &= T_K + P_I/g_{inter} \\ &= T_{amb} + (P_K + P_I)/g_{hs} + P_I/g_{inter} \end{aligned} \quad (2)$$

where T_K and T_I are the temperatures of Cores K and I, T_{amb} is the ambient temperature, P_K and P_I are the power consumptions of Cores K and I. In addition to Core I thermally dominating Core K, it also has a higher total resistance to the ambient, i.e., it has a lower *cooling efficiency*. As a result, a unit of power consumption on Core I will have at least as great an impact on temperature as a unit of power consumption on Core J or K.

Thermal Coupling: The thermal conductance between J and K (g_{intra}) is approximately 0.41 W/K. Heat can flow between Cores J and K. As a result, the power consumption of one can influence the temperature of the other. However, this thermal coupling is relatively minor compared to that between vertically-aligned cores. The thermal conductance between Cores I and K (g_{inter}) is approximately $16 \times g_{intra}$. The large interface area between Cores I and K results in a high thermal conductance, despite the interposed high thermal resistivity (but thin, and therefore low resistance) 10 µm polyimide bonding layer.

Summary and Open Questions: At this point, we can draw some qualitative conclusions. The temperatures of vertically-aligned cores are highly correlated, relative to the temperatures of horizontally-adjacent cores. Cores farther from the heatsink have higher temperatures than their neighbors closer to the heatsink. In addition, the temperature impact of a unit of power dissipation will be at least as high for Core I as for Cores J and K, due to their differing thermal conductances to the ambient. However, a few questions remain:

1) How can we use this knowledge of thermal environment heterogeneity to guide the development of a CMP thermal management algorithm? and
2) What is the impact of the power consumption of each core upon all other cores in the system?

We will now introduce a general analytical framework that answers these questions.

III. 3D CMP Thermal Management

In this section, we investigate the 3D CMP run-time thermal management problem and propose efficient management techniques. Given a 3D CMP with N on-chip processor cores, our goal is to maximize the CMP throughput under a constraint on peak temperature. CMP throughput is defined as the total number of instructions executed by the CMP per second.

$$CMP\_IPS = \sum_{i=0}^{N-1} IPC_i \times f_i \quad (3)$$

where IPC_i and f_i are the run-time instructions per cycle and frequency of Core i.

Run-time thermal safety requires that

$$\forall_{i=0}^{N-1}\, T_i \leq T_{MAX} \quad (4)$$

i.e., the temperature of each processor core cannot exceed the maximum safe temperature: T_{MAX}.

In the following sections, we analyze the thermal management problem for 3D CMPs and determine the policies necessary for performance optimization given a temperature constraint. This study will be used to guide the development of our run-time thermal management techniques.

A. Conditions and Guidelines Required for Optimal 3D CMP Thermal Management

This section presents performance optimization guidelines. The central theme is to optimize the performance of CMP cores under a constraint on peak temperature during workload assignment and power–thermal budgeting.

Observation: *To maximize CMP throughput, processor cores should operate at different voltages and frequencies due to heterogeneous processor core thermal characteristics and heterogeneous run-time workloads.*

As described in Figure II-A, processor cores in a 3D CMP are thermally correlated. The temperature of each Core i, is affected by the power consumptions of all cores, as follows:

$$T_i = \sum_{j=0}^{N-1} \zeta_{i,j} \times p_j \leq T_{MAX} \quad (5)$$

where T_i is the temperature of processor Core i; $\zeta_{i,j}, \{i,j\} \in [0, N-1]$ is an inter-core thermal impact coefficient, which indicates the impact of a unit power consumption of Core j on the temperature of Core i; p_j is Core j's power consumption; and N is the number of processor cores of the CMP.

We would like to guide migration of tasks among cores, and budget power to cores, in order to optimize CMP throughput under a temperature constraint. To facilitate developing the necessary guidelines, we introduce the concept of thermal impact per performance gain, TIP:

$$TIP^{f}_{i,j} = \frac{dT_i}{df_j},\ TIP^{IPC}_{i,j} = \frac{dT_i}{dIPC_j} \tag{6}$$

$TIP_{i,j}$ indicates the thermal impact on processor Core i due to the increase in Core j's performance, by either increasing its frequency and voltage, and/or assigning a high IPC job to this core. Intuitively, TIP is the thermal cost per unit increase in processor core performance. It can be viewed as the inverse of a core's *thermal efficiency*. Subject to a temperature bound, maximizing CMP performance thus requires that all the processor cores achieve the same thermal impact per performance improvement on the maximum-temperature core, i.e.,

$$TIP^{f,IPC}_{i,0} \equiv TIP^{f,IPC}_{i,1} \equiv \cdots \equiv TIP^{f,IPC}_{i,N-1} \tag{7}$$

Note that the impact on T_i due to the power consumption of core j is $\zeta_{i,j}P_j$. Given that dynamic power consumption, $P_j = \xi_j V_j^2 f_j$ (where V_j and f_j are the supply voltage and frequency of Core j), $V_j \propto f_j^{\beta}$, and $\beta \approx 1$ [30]; ξ_j is Core j's run-time switching activity multiplied the capacitance of the switched nodes (which is approximately linearly-proportional to the IPC of the job running in Core j), then

$$\begin{aligned} \zeta_{i,0} f_0^{2\beta+1} &\equiv \zeta_{i,1} f_1^{2\beta+1} \equiv \cdots \equiv \zeta_{i,N-1} f_{N-1}^{2\beta+1} \\ \zeta_{i,0}\xi_0 f_0^{2\beta} &\equiv \zeta_{i,1}\xi_1 f_1^{2\beta} \equiv \cdots \equiv \zeta_{i,N-1}\xi_{N-1} f_{N-1}^{2\beta} \end{aligned} \tag{8}$$

This result indicates that processor cores with heterogeneous power and thermal characteristics, i.e., different power–thermal impact coefficients, $\zeta_{i,j}$, running jobs with different IPCs should be clocked at different frequencies. A similar conclusion can be drawn when both dynamic and leakage power consumption are considered.

As shown in Section II-A, the inter-layer and intra-layer thermal characteristics of 3D CMPs show distinct differences. This leads to different thermal management policies for inter-layer and intra-layer processor cores.

1) Inter-Layer Power–Thermal Budgeting and Workload Assignment: Inter-layer processor cores have heterogeneous thermal characteristics. In addition, vertically-aligned cores have strongly-correlated temperatures. We now present heterogeneity-aware guidelines for power–thermal budgeting and workload assignment among vertically-aligned cores.

Guideline I: *To maximize CMP throughput, the thermal efficiencies of vertically-aligned processor cores should be optimized under the thermal constraint, i.e., the voltage and frequency assignment among vertically-aligned processor cores should follow Equations 5–8.*

Guideline II: *Given jobs with different IPCs, the maximum CMP throughput can only be achieved by maximizing the IPC heterogeneity during workload distribution. To maximize throughput, jobs with higher IPCs should be assigned to cores with higher thermal efficiencies.*

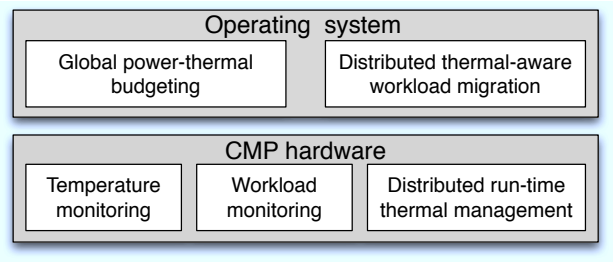

Fig. 3. ThermOS: 3D CMP run-time thermal management.

2) Intra-Layer Power–Thermal Budgeting: Intra-layer cores have mostly-homogeneous thermal characteristics with almost identical cooling efficiencies (see Section II-A), i.e., $\zeta_{i,i} \approx \zeta_{j,j}$, when Core i and Core j are in the same layer. In addition, the inter-core thermal impact is significantly lower than the self power–thermal impact of each core, i.e., $\zeta_{i,i} \gg \zeta_{i,j}$, when $i \neq j$. We therefore propose the following policy for intra-layer power–thermal budgeting and workload assignment.

Guideline III: *To maximize aggregate CMP frequency or instruction throughput, power–thermal budget and workload should be balanced among intra-layer processor cores.*

B. ThermOS: 3D CMP Thermal Management

Based on the thermal management guidelines presented in Section III-A, we have developed ThermOS, a unified hardware and OS thermal management solution to maximize thermally-safe 3D CMP performance. As shown in Figure 3 and Table I, ThermOS consists of hardware-based temperature–workload monitoring and distributed run-time thermal management built into a 3D CMP microarchitecture, as well as a temperature-aware Linux kernel equipped for global power–thermal budgeting and distributed temperature-aware workload migration. ThermOS is a proactive, continuously-engaged solution designed to handle 3D CMP power–thermal heterogeneities, distribute run-time workload, and manage the limited power–thermal budget to optimize performance under temperature constraint. ThermOS is built upon the Linux 2.6.8 kernel. It has an $\mathcal{O}(1)$ time complexity scheduler. Our temperature-aware scheduling algorithm maintains the same time complexity. Table I summarizes the proposed offline and run-time software and hardware management techniques.

1) Temperature Monitoring: ThermOS gathers CMP thermal profiles at run-time, which are used to guide temperature-aware workload migration as well as power–thermal budgeting. Either thermal sensors or online thermal analysis may be used for on-line temperature monitoring. Thermal sensors have been widely used in high-performance microprocessors [31], [1]. Efficient software-based online thermal analysis techniques have also been developed [19].

2) Workload Monitoring: In addition to CMP thermal profile, ThermOS gathers run-time performance and power characteristics to guide job migration as well as power–thermal budgeting. A processor core's *activity factor* is a function of the capacitances of its functional units and the corresponding

TABLE I
THERMOS IMPLEMENTATION

Offline computation			Given the activity factor range of on-chip processor core, derive a look-up table that contains the optimal voltages and frequencies yielded by Equations 8–11.
Online	OS	rebalance_tick()	Invoke cluster_opt() and group_opt() at the beginning of each workload migration time interval (every 20 ms).
		cluster_opt()	Conduct inter-layer migration according to Guideline II.
		group_opt()	Conduct intra-layer migration according to Guideline III.
		scheduler_tick()	1) Monitor the activity factors of run-time processes using hardware performance counters. 2) Determine the global power–thermal budgeting using run-time table lookup.
	Hardware	Local DVFS	Proactive distributed DVFS based on global guidance and local variation.
		Local clock	Reactive distributed clock throttling to guarantee thermal safety.

run-time activity factors resulting from its workload. Most modern processors provide hardware performance counters for monitoring specific events [1], [32]. These performance counters can be used to inform accurate and efficient regression-based run-time performance and power models [33], [34]. ThermOS uses this technique for linear regression estimation of run-time processor core activity factors. The model was developed offline and integrated with the OS. During execution, each processor core's hardware performance counter values are gathered periodically when triggered by OS timer interrupts (every 1 ms in Linux 2.6.8 kernel). These performance counter values are used for run-time workload activity and IPC estimation.

3) Distributed Temperature-Aware Workload Migration: ThermOS contains a distributed online workload migration technique to support performance optimization. The proposed technique follows the guidelines presented in Section III-A and carefully handles 3D CMP inter-layer thermal heterogeneity and run-time workload heterogeneity. ThermOS uses a distributed approach that swaps jobs with high IPCs to processor cores with higher thermal efficiencies.

Consider two vertically-adjacent processor cores: Core I and Core K. Assume Core K has a higher thermal efficiency than Core I. To optimize instruction throughput, ThermOS compares the jobs stored in each processor core's job queue. It first identifies the lowest-IPC job ($IPCMIN_K$) on core K and the highest-IPC job ($IPCMAX_I$) on Core I. If $IPCMIN_K < IPCMAX_I$, ThermOS swaps the corresponding jobs. Intra-layer thermal heterogeneity and thermal correlation are small. Therefore, ThermOS balances the intra-layer IPC distribution to optimize instruction throughput. Average IPCs of jobs on horizontally-adjacent cores are compared. If appropriate, they are swapped to further balance the distribution. The proposed distributed temperature-aware workload migration technique has been integrated within the default Linux kernel workload balancing policy. In the current implementation, workload migration occurs every 20 ms.

4) Global Power–Thermal Budgeting: ThermOS dynamically adjusts the power–thermal budgets of processor cores to optimize 3D CMP performance. Following the guidelines in Section III-A, ThermOS balances the power–thermal budget assignment among processor cores in the same layer. Equations 5–8 are used to guide inter-layer power–thermal budgeting. The leakage–temperature dependency introduces temperature variables on both sides of Equation 7. Solving this equation requires numerical iteration and detailed chip-package thermal analysis, which are computationally intensive. To minimize run-time overhead, we have developed hybrid offline/online budgeting technique.

Given the switching activity (or IPC) range of the workload, the optimal voltage and frequency settings for vertically-aligned processor cores are pre-computed. The offline component of the budgeting algorithm is iterative. During each iteration, based on the IPC and the switching activity of each processor core, Equations 5–8 are used to determine the optimal processor core power–thermal budgets. Thermal analysis is then used to estimate the 3D CMP thermal profile and update the leakage power profile estimate. This process iterates until the chip-package thermal profile converges, subject to feedback from temperature-dependent leakage power consumption. The final voltage and frequency configurations are stored in a look-up table for efficient use during online power–thermal budgeting. In ThermOS, run-time power–thermal budgeting is implemented in the Linux kernel and invoked periodically. Periods ranging from 1 ms to 100 ms are currently supported.

5) Distributed Run-Time Thermal Management: ThermOS uses distributed run-time thermal management to honor the power and thermal budgets described in Section III-B4 and adhere to a temperature constraint. Periodically, each processor core adjusts its voltage and frequency based on its assigned power–thermal budget. However, transient variations may not be immediately detected by the OS. In order to honor the temperature constraint, ThermOS uses local dynamic voltage and frequency scaling (DVFS) and clock throttling to react to transient variation with lower latency than global power–thermal budgeting. DVFS has a higher response latency than clock throttling; for modern high-performance microprocessors, the voltage transition rate is in the range of 10 mV/µs [35]. Clock throttling, on the other hand, has low latency. However, DVFS has less performance impact per unit power reduction than clock throttling, thanks to the superlinear dependence of power on voltage. Note that most modern high-performance processors already support DVFS. We are proposing to use this existing DVFS hardware to the best effect. In ThermOS, local DVFS continuously tracks temperature changes and clock throttling is used as a final defense to guarantee thermal safety.

IV. EXPERIMENTAL RESULTS

This section evaluates ThermOS, the proposed run-time thermal management solution for 3D CMPs. We use the M5 full system simulator [26], into which we have integrated a power model, thermal model, and ThermOS. We use a set of multithreaded and multiprogrammed benchmarks from SPEC2000, Media Bench, ALPBench, and SPLASH2, which are shown in Table II. These benchmarks are further parti-

TABLE II
BENCHMARK CHARACTERISTICS

Group	Name	Avg. IPC	Avg. Pow. (W)	Max. T	Max. δT
SPEC High IPC	gcc	3.36	14.67	64.88	0.20
	applu	3.13	14.37	65.64	0.12
	gzip	2.78	13.34	63.49	0.34
	mgrid	2.58	13.66	61.84	0.31
SPEC Low IPC	twolf	1.58	11.33	64.30	0.19
	parser	1.55	10.41	60.70	0.28
	vpr	1.47	10.63	60.43	0.29
	mcf	1.25	10.91	63.79	0.25
Media High IPC	gsmenc	3.10	13.50	63.38	0.09
	jpegdec	2.72	13.42	65.89	0.13
Media Low IPC	g721enc	1.94	11.91	61.39	0.08
Multithreaded (two threads)	MPGenc	2.95	14.34	68.78	0.20
	Sphinx3	1.13	9.93	61.68	0.02
	cholesky	2.83	14.27	70.57	0.32
	lu	2.26	12.10	66.97	0.08
	radix	0.84	5.81	57.17	0.28
	water-nsquared	1.85	11.99	65.32	0.12
	water-spatial	1.74	10.57	62.35	0.08

TABLE III
BENCHMARK SUITES

Multiprogrammed test setups

Group	Filename	Clusters	Benchmarks
SPEC	hv-hipc	High T var., high IPC	gzip, mgrid
	lv-hipc	Low T var., high IPC	applu, gcc
	hv-lipc	High T var., low IPC	parser, vpr
	lv-lipc	Low T var., low IPC	twolf, mcf
	hv-mipc1	High T var., mixed IPC	gzip, parser
	hv-mipc2	High T var., mixed IPC	mgrid, vpr
	lv-mipc1	Low T var., mixed IPC	applu, mcf
	lv-mipc2	Low T var., mixed IPC	gcc, twolf
Media	media-hipc	High IPC	jpegdec, gsmenc
	media-mipc	Mixed IPC	gsmenc, g721enc

Multithreaded test setups

MPGenc, sphinx3, cholesky, lu, radix, water-nsquared, water-spatial

tioned into 17 test setups (see Table III) using two benchmark-specific metrics, IPC and expected temperature variation.

The ThermOS run-time thermal management algorithms are implemented within the Linux 2.6.8 kernel. We made two main changes to the kernel:

- *Performance-counter based power modeling:* We enable OS-level power estimation using performance counters. Hardware event counters of the sort typical for modern processors were added to M5. A regression-based power model was added to the OS [33].
- *Power–thermal budgeting, task migration, and thermal management:* The proposed power–thermal budgeting and temperature-aware task migration techniques were implemented in the Linux kernel. We modified M5 to support kernel control of DVFS and clock throttling temperature monitoring through privileged machine registers.

A. Comparison of ThermOS With Alternatives

In this section, we first contrast ThermOS with solutions used in existing processors. Then we provide a detailed quantitative comparison with a state-of-the-art continuously-engaged thermal management technique. The following experiments use 85 °C as a predefined thermal constraint.

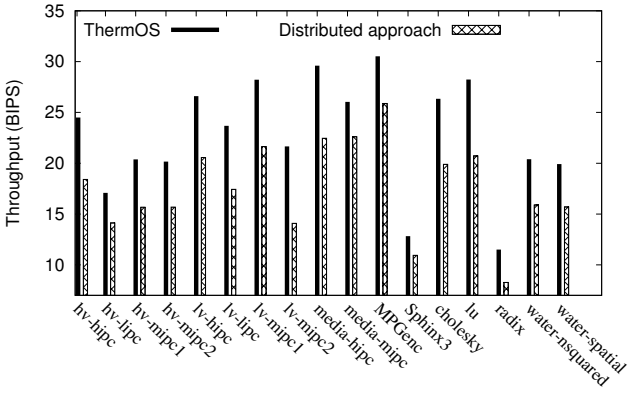

Fig. 4. Comparison of ThermOS and distributed approach [17].

Most thermal management techniques used in practice react to emergencies instead of being continuously engaged. They detect dangerously-high temperatures and reduce power consumption, generally via hardware clock throttling. Such solutions are adequate when temperatures approach their limits only very rarely. However, high power densities and constraints on cooling costs require proactive thermal management. Some researchers have moved in this direction.

Donald and Martonosi [17] proposed a distributed continuously-engaged thermal management technique for 2D CMPs. Their approach is based on closed-loop control theory, and continuously adjusts the voltage and frequency of each processor core to maintain safe temperatures. Each core has its own controller and the controllers act independently, without knowledge of the conditions of other cores. This permits significantly better performance than reactive approaches because DVFS can generally reduce power consumption by the same amount as clock throttling with a smaller performance penalty. In fact, their results indicate that, compared with a stop-go based thermal control policy, distributed DVFS improves throughput by 2.5×. However, independent local control has limitations. The power consumed in one processor can impact the temperatures of other processors in nonuniform ways. As a result, continuously-engaged global control can permit better performance than continuously-engaged local control. This is especially true for 3D architectures, in which the power consumption of a particular processor core has great impact on the temperature of vertically-aligned cores and relatively less impact on other cores.

ThermOS uses continuously-engaged, distributed global/local control to maximize performance given a temperature constraint. It supports both 3D and 2D architectures. It has two primary differences with state-of-the-art temperature control techniques. First, it uses global power budgeting that takes into account the thermal interaction between processor cores. Second, it directs temperature-aware workload migration of threads among processor cores.

Figure 4 shows the 3D CMP run-time instruction throughputs in billion of instructions per second (BIPS), achieved by ThermOS and Donald's and Martonosi's approach. Compared to the distributed local approach, ThermOS improves instruction throughput by 29.84% on average (ranging from 15.22% to 53.79%). This can be explained as follows. In 3D CMPs, the

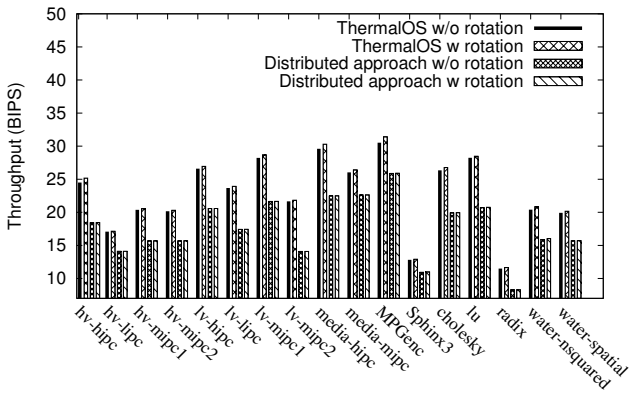

Fig. 5. Impact of floorplan rotation.

strong thermal correlation among inter-layer vertically-aligned processor cores has significant impact on the temperature of the processor layer farthest from the heat sink. Using the proposed power–thermal budgeting and temperature-aware workload migration techniques, ThermOS determines appropriate power budgets for each group of vertically-aligned processor cores. In addition, it uses workload migration and DVFS to optimize the power–thermal efficiency of each processor core. Together, these techniques maximize overall throughput. Donald's and Martonosi's work, on the other hand, is a distributed, processor-local technique. Using this technique, each processor core regulates its power and performance to ensure local thermal safety without considering the thermal impact on neighboring cores. As a result, vertically-aligned processor cores are unable to collaboratively share the power–thermal budget, which can reduce CMP performance. In other words, when a distributed, local management technique is used, power consumption on processor cores near the heatsink can push processor cores farther from the heatsink to their thermal limits.

B. Robustness to Changes in 3D Integration

In order to show the robustness of ThermOS to variation in 3D integration style, we evaluated the performance improvement when used for CMPs based on front-to-back and front-to-front wafer integration. We simulated the proposed technique and Donald's and Martonosi's distributed local approach [17] for both integration styles using all benchmark mixes shown in Table III. The average CMP instruction throughput improvement was 29.84% for front-to-back integration and 23.77% for front-to-front integration. For all combinations of benchmarks and packages, the instruction throughput improvements were greater than 7%. We can conclude that ThermOS permits substantial improvements in performance over Donald's and Martonosi's distributed local technique for different 3D integration styles.

C. Interaction with 3D CMP Floorplan Optimization

This experiment evaluates ThermOS for 3D CMPs with different floorplans. CMP thermal profile is strongly influenced by on-die power distribution. In 3D CMPs, inter-layer vertically-aligned processor cores have strong thermal correlation. If all cores have identical floorplans, functional units with high power densities are vertically-aligned, potentially creating local thermal hotspots. Intelligent inter-layer floorplan arrangement can potentially balance inter-layer power profile and minimize chip peak temperature. Using the three-layer 3D CMP setup with processor core layers and one L2 cache layer, detailed thermal analysis shows that, by rotating the floorplan of top-layer processor cores by 180 degrees, chip power profile is more balanced, intra-core local hotspots are minimized, and chip peak temperature is reduced by 1.99 °C on average and 4.24 °C maximum among the multiprogramming and multithreading benchmarks. Figure 5 compares ThermOS and the baseline distributed technique, with and without floorplan rotation. It shows that both run-time techniques can leverage the temperature reduction offered by floorplan rotation and achieve higher throughput under the same temperature constraint. In addition, ThermOS consistently outperforms the distributed technique by 31.45% and 29.84% on average with and without floorplan rotation, respectively.

D. Interactions Between Thermal Management and Reliability

As described in Section I, high or varying temperatures can result in increased wear due to lifetime fault processes, e.g., electromigration. This implies that a reduction in average temperature due to more effective thermal management can increase integrated circuit lifetime. There is even the potential to estimate this effect [36] when designing and selecting parameters for a thermal management algorithm, thereby producing a thermal management technique that maximizes performance under a constraint on desired integrated circuit lifetime.

Variation in temperature can also result in transient timing faults because charge carrier mobility reduces with increased temperature. This is a result of high temperatures increasing the frequency of interaction among charge carrier and phonons in the atomic lattice of semiconductors and metal, thereby decreasing the charge carrier mean free path, i.e., decreasing the conductance. This effect can result in timing violations due to run-time changes in temperature. It is generally compensated for by constraining the temperature of the integrated circuit to a safe range by dynamically adjusting the power consumption of the processor at risk. When selecting appropriate power management techniques for use within thermal management, there is tension between power–performance efficiency and latency. Some power control techniques, such as DVFS, have good power reduction to performance reduction ratios, but have high reaction latencies, e.g., 10 mV/µs [35]. Others, such as clock throttling, have relatively-poor power reduction to performance reduction ratios, but can reduce power consumption almost immediately.

ThermOS uses multiple power control techniques that trade off power–performance efficiency and reaction latency. We found that the vast majority of potential thermal emergencies result from gradual changes in temperature: DVFS can protect against them. However, in a few cases temperature changes so rapidly that a low-latency power control technique (clock throttling) must be used. Fortunately, these cases are so rare that the poor power–performance efficiency of clock throttling has negligible impact on the overall performance of ThermOS.

V. Conclusions

3D integration has the potential to significantly improve performance and integration density. However, it will increase power density, thereby increasing the importance of using continuously-engaged thermal management techniques. It will also increase the heterogeneity in thermal interaction among processor cores. This requires careful consideration during thermal management policy design.

We have developed a mathematical formulation for optimizing workload assignment, power–thermal budgeting, and voltage mode selection for 3D CMP thermal management. This formulation has been used to develop ThermOS, a continuously-engaged hardware–software thermal management solution for 3D CMPs. The proposed solution has been implemented within the Linux kernel and evaluated using full-system 3D CMP and OS simulation. ThermOS provides a near-optimal thermal management solution to maximize the thermally-safe performance of future 3D and 2D CMPs.

References

[1] R. Kalla, B. Sinharoy, and J. M. Tendler, "IBM Power5 chip: a dual-core multithreaded processor," *IEEE Micro*, vol. 24, no. 2, pp. 40–47, 2004.

[2] P. Kongetira, K. Aingaran, and K. Olukotun, "Niagara: A 32-way multithreaded SPARC processor," *IEEE Micro*, vol. 25, no. 2, pp. 21–29, 2005.

[3] "AMD multi-core white paper," http://www.amd.com.

[4] "Intel multi-core processor architecture," http://www.intel.com.

[5] M. B. Taylor, et al., "Evaluation of the raw microprocessor: An exposed-wire-delay architecture for ILP and streams," in *Proc. Int. Symp. Computer Architecture*, June 2004, pp. 2–13.

[6] K. Sankaralingam, R. Nagarajan, H. Liu, J. Huh, C. K. Kim, D. Burger, S. W. Keckler, and C. R. Moore, "Exploiting ILP, TLP, and DLP using polymorphism in the TRIPS architecture," in *Proc. Int. Symp. Computer Architecture*, June 2003, pp. 422–433.

[7] S. Vangal, et al., "An 80-tile 1.28TFLOPS networks-on-chip in 65nm CMOS," in *Proc. Int. Solid-State Circuits Conf.*, Feb. 2007, pp. 98–100.

[8] V. Agarwal, M. Hrisikesh, S. Keckler, and D. Burger, "Clock rate vs. IPC: The end of the road for conventional microarchitectures," in *Proc. Int. Symp. Computer Architecture*, June 2000, pp. 276–283.

[9] A. W. Topol, et al., "Three-dimensional integrated circuits," *IBM J. Research and Development*, vol. 4, pp. 491–506, 2006.

[10] B. Black, et al., "Die stacking (3D) microarchitecture," in *Proc. Int. Symp. Microarchitecture*, Dec. 2006, pp. 469–479.

[11] Samsung, http://www.samsung.com/.

[12] Tezzaron, http://www.tezzaron.com/technology/FaStack.htm.

[13] F. Li, C. Nicopoulos, T. Richardson, Y. Xie, V. Narayanan, and M. Kandemir, "Design and management of 3D chip multiprocessors using network-in-memory," in *Proc. Int. Symp. Computer Architecture*, June 2006, pp. 130–141.

[14] T. Kgil, S. D'Souza, A. Saidi, N. Binkert, R. Dreslinski, T. Mudge, S. Reinhardt, and K. Flautner, "PicoServer: using 3D stacking technology to enable a compact energy efficient chip multiprocessor," in *Proc. Int. Conf. Architectural Support for Programming Languages and Operating Systems*, Oct. 2006, pp. 117–128.

[15] J. Kim, C. Nicopoulos, D. Park, R. Das, Y. Xie, V. Narayanan, M. S. Yousif, and C. R. Das, "A novel dimensionally-decomposed router for on-chip communication in 3D architectures," in *Proc. Int. Symp. Computer Architecture*, June 2007, pp. 138–149.

[16] Y. Li, D. Brooks, Z. Hu, and K. Skadron, "Performance, energy, and thermal considerations for SMT and CMP architectures," in *Proc. Int. Symp. Computer Architecture*, Feb. 2005, pp. 71–82.

[17] J. Donald and M. Martonosi, "Techniques for multicore thermal management: Classification and new exploration," in *Proc. Int. Symp. Computer Architecture*, June 2006, pp. 78–88.

[18] D. Brooks and M. Martonosi, "Dynamic thermal management for high-performance microprocessors," in *Proc. Int. Symp. High-Performance Computer Architecture*, Jan. 2001, pp. 171–182.

[19] K. Skadron, M. R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan, "Temperature-aware microarchitecture," in *Proc. Int. Symp. Computer Architecture*, June 2003, pp. 2–13.

[20] K. Puttaswamy and G. H. Loh, "Thermal analysis of a 3d die-stacked high-performance microprocessor," in *Proc. Great Lakes Symp. VLSI*, May 2006, pp. 19–24.

[21] K. Puttaswamy and G. H. Loh, "Thermal herding: Microarchitecture techniques for controlling hotspots in high-performance 3d-integrated processors," in *Proc. Int. Symp. High-Performance Computer Architecture*, Feb. 2007, pp. 193–204.

[22] G. M. Link and N. Vijaykrishnan, "Thermal trends in emerging technologies," in *Proc. Int. Symp. Quality of Electronic Design*, Mar. 2006, pp. 625–632.

[23] M. D. Powell, M. Gomaa, and T. N. Vijaykumar, "Heat-and-run: Leveraging SMT and CMP to manage power density through the operating system," in *Proc. Int. Conf. Architectural Support for Programming Languages and Operating Systems*, Nov. 2004, pp. 260–270.

[24] J.McGregor, "x86 power and thermal management," in *Microprocessor Report*, Dec. 2004, pp. 1–8.

[25] A. Mallik, J. Cosgrove, R. P. Dick, G. Memik, and P. Dinda, "PICSEL: Measuring user-perceived performance to control dynamic frequency scaling," in *Proc. Int. Conf. Architectural Support for Programming Languages and Operating Systems*, Mar. 2008.

[26] N. L. Binkert, R. G. Dreslinski, L. R. Hsu, K. T. Lim, A. G. Saidi, and S. K. Reinhardt, "The M5 simulator: Modeling networked systems," *Proc. Int. Symp. Microarchitecture*, vol. 26, no. 4, pp. 52–60, 2006.

[27] COMSOL Multiphysics. COMSOL, Inc. http://www.comsol.com/products/multiphysics.

[28] ANSYS. http://www.ansys.com.

[29] Y. Yang, C. Zhu, Z. P. Gu, L. Shang, and R. P. Dick, "Adaptive multi-domain thermal modeling and analysis for integrated circuit synthesis and design," in *Proc. Int. Conf. Computer-Aided Design*, Nov. 2006, pp. 575–582.

[30] K. A. Bowman, B. L. Austin, J. C. Eble, X. Tang, and J. D. Meindl, "A physical alpha-power law MOSFET model," *IEEE J. Solid-State Circuits*, vol. 34, pp. 1410–1414, Oct. 1999.

[31] D. Pham, et al., "The design and implementation of a first-generation CELL processor," in *Proc. Int. Solid-State Circuits Conf.*, Feb. 2007, pp. 49–52.

[32] R. Sprunt, "Pentium 4 performance-monitoring features," *IEEE Micro*, vol. 22, no. 4, pp. 72–82, 2002.

[33] C. Isci and M. Martonosi, "Runtime power monitoring in high-end processors: Methodology and empirical data," in *Proc. Int. Symp. Microarchitecture*, Dec. 2003, pp. 93–104.

[34] A. Kumar, L. Shang, L.-S. Peh, and N. K. Jha, "HybDTM: a coordinated hardware-software approach for dynamic thermal management," in *Proc. Design Automation Conf.*, July 2006, pp. 548–553.

[35] C. Isci, A. Buyuktosunoglu, C.-Y. Cher, P. Bose, and M. Martonosi, "An analysis of efficient multi-core global power management policies: Maximizing performance for a given power budget," in *Proc. Int. Symp. Microarchitecture*, Dec. 2006, pp. 347–358.

[36] C. Zhu, Z. P. Gu, R. P. Dick, and L. Shang, "Reliable multiprocessor system-on-chip synthesis," in *Proc. Int. Conf. Hardware/Software Codesign and System Synthesis*, Oct. 2007, pp. 239–244.