Modeling and Analysis of Phase Change Materials for Efficient Thermal Management

Fulya Kaplan†, Charlie De Vivero†, Samuel Howes†, Manish Arora*§, Houman Homayoun‡,
Wayne Burleson*, Dean Tullsen§ and Ayse K. Coskun†

†ECE Department, Boston University, Boston, MA – {fkaplan3, devivero, showes, acoskun}@bu.edu
*Advanced Micro Devices, Inc., Boxborough, MA – {manish.arora, wayne.burleson}@amd.com
‡ECE Department, George Mason University, Fairfax, VA – hhomayou@gmu.edu
§CSE Department, University of California, San Diego, CA – tullsen@cs.ucsd.edu

***Abstract*—Direct placement of Phase Change Materials (PCMs) on the chip has been recently explored as a passive temperature management solution. PCMs provide the ability to store large amounts of heat at a close-to-constant temperature during the phase change (solid to liquid and vice versa). This latent heat capacity can be used to provide higher performance while reducing hot spots. Detailed modeling of the phase change behavior is essential for the design and evaluation of systems with PCM. This paper proposes an accurate phase change model that is integrated into the commonly used thermal simulation tool, HotSpot. It also provides validation of the proposed model by carrying out computational fluid dynamics (CFD) simulations using COMSOL Multiphysics®. This paper also explores the impact of PCM properties on the thermal profile of a processor, and demonstrates that PCM material choices can affect peak temperatures by up to $20.1^{o}C$. Experimental results show that dynamic policy decisions change dramatically when using the proposed detailed phase change model, as prior simpler PCM models can substantially over/under-estimate temperature and PCM melting duration. The proposed model helps design more effective dynamic management policies and enables realistic evaluation of systems with PCM.**

I. INTRODUCTION

Continued technology scaling has led to power density increases, making temperature management a critical aspect of processor system design today. Effective management of processor temperature is necessary as higher temperatures increase leakage power, reduce performance, and degrade chip reliability. In order to reduce temperature, active and passive cooling techniques are used, along with runtime power and thermal management techniques. Active cooling techniques include forced air cooling with fans, liquid cooling through microchannels [1], heat pipes in fanless systems [2] and thermoelectric cooling [3]. On the other hand, passive cooling such as using a heat sink is an attractive option for low-power systems. However, size constraints make the use of heat sinks difficult in embedded or mobile systems.

Recently, the use of **Phase Change Materials (PCMs)**[1] has been proposed as a passive cooling technique [4], [5], [6]. PCMs are compounds that store and release thermal energy during the process of melting/freezing. PCM-based cooling uses the change of the state of the material, e.g., solid to liquid, to provide a thermal heat buffer and absorb the heat generated on the die. This capacity to absorb heat during phase change is referred to as the latent heat capacity. PCM-based cooling is attractive for two reasons. (1) The latent heat capacity per unit volume can be very high (hundreds of joules per cm^{3}), allowing for large amounts of heat to be absorbed in a small volume. (2) Temperature is relatively constant during phase change, allowing the system to have much smoother temperature profiles. Thus, PCMs can reduce or eliminate the need for active cooling solutions or thermal throttling effects by enabling more cost-efficient temperature management.

The current research landscape of the application of PCMs to computing systems [4], [5], [6] centers around the ideas of placing these materials on top of the chip and developing processor power management techniques that are aware of the use of PCMs in the system. Existing work assumes material choices with near-optimal thermal and physical properties [4]. However, the choice of material with desirable properties (in terms of the heat storage capacity and conductivity) or the volume of the material to be placed on the chip may not always be straightforward. Thus, a design space exploration is needed to make appropriate material selections to maximize the benefits of PCM on the chip. In addition, a detailed transient analysis of systems with PCM is necessary to enable design of efficient runtime management strategies. Such an evaluation is limited by the lack of tools and models that accurately evaluate the impact of PCMs for realistic systems and benchmarks. The models used in recent work are either coarse-grained or strongly rely on specific assumptions [4], [6]. Those models fail to model more complex behavior, such as local melting around hot cores while other parts of the PCM are still solid, allowing portions of the PCM to heat up while others are absorbing heat. Missing this phenomenon can result in significant under- or over-estimation of temperature.

This paper addresses this gap and introduces a detailed thermal model for PCM on chip, and then uses this model to explore the PCM design space and the impact of using temperature management policies on systems with PCM. Thus, this work advances the latest PCM research by exploring the design space and runtime behavior at a finer spatial and temporal granularity. Our main contributions are as follows:

- We propose a fine-grained phase change model to account

[1] PCM has also been used when referring to phase change memory in recent literature. This paper focuses on using phase change materials as thermal buffers, and not on memories built with PCMs.

for the thermal behavior of PCM. We integrate the proposed model into the commonly used HotSpot temperature simulator [7]. We validate the accuracy of our model by comparing its results against COMSOL Multiphysics CFD module simulations (hereafter referred to as COMSOL) [8].

- We leverage the PCM thermal model to investigate the effect of PCM thermal properties on the CPU temperature profile. We demonstrate that across different PCM configurations, the resulting peak and average temperatures differ by 20.1^oC and 11.7^oC, respectively.
- We compare our model against the PCM model used in prior work [6] and show that our detailed model captures non-uniform melting of the PCM layer, leading to more realistic evaluation of dynamic thermal management policies. We also show that assuming uniform melting leads to 12.5x average difference in the number of throttling instances.

The rest of the paper starts with a discussion of prior work. Section III explains our phase change model and provides the validation results against COMSOL. Section IV provides the experimental methodology. We present the experimental results in Section V and conclude in Section VI.

II. Related Work

This section discusses the prior work on (1) processor thermal management, (2) using PCM as heat storage units, and (3) leveraging PCM in the context of computational sprinting.

Thermal management in multicore systems is a widely studied topic. Some prior work proposes thermally-aware dynamic voltage and frequency scaling (DVFS) techniques [9], [10], [11]. Bao et al. introduce an online DVFS technique, which shows that the allowed frequency for a given voltage setting depends on the operating temperature, and they use this frequency slack to improve energy efficiency [9]. Bartolini et al. propose a distributed model-predictive controller to minimize the energy consumption of a large multicore platform. Their solution decides on the optimal frequency for each node and saves energy while preserving a tolerable performance loss, as well as obeying thermal constraints [10]. Coskun et al. propose a hybrid technique combining workload allocation and DVFS to increase lifetime of a multicore system with temperature, energy, and performance considerations [11].

Recent studies explore the benefits of PCM in electronic thermal management [4], [5], [12], [13], [14], [15]. Tan et al. use CFD simulation for thermal management of a mobile phone with a PCM filled heat storage unit [13]. Alawadhi et al. study the effectiveness of a thermal control unit composed of a PCM and a thermal conductivity enhancer on a portable electronic device [12]. Yoo et al. investigate the energy savings of using PCM with a heat sink as an alternative to a fan-cooled heat sink [14]. Stupar et al. propose a hybrid air-cooled heat sink containing PCM for high peak load, low duty cycle applications [15].

PCM has been recently proposed to enhance the efficiency of performance boosting policies. Raghavan et al. introduce the concept of *computational sprinting*, which exploits the PCM by exceeding the thermal design power (TDP) budget for short bursts, and investigate the benefits of using PCM in extending the sprinting duration [4]. Their work considers high intensity sprinting, which is turning on all available cores and running them at the highest voltage/frequency setting until a temperature threshold is reached. This approach is a threshold based on/off policy. Following up on this work, they analyze the feasibility of sprinting using a hardware/software test bed [5]. They also introduce the concept of sprint pacing and propose an adaptive policy for selecting the sprint intensity. Tilli et al. propose safe computational re-sprinting for periodic hard deadline tasks [6]. They provide a control policy that keeps the PCM latent heat capacity at a particular level to guarantee sprinting at full power for a known duration.

Prior work models the phase change behavior using various approaches. Sridhar et al. propose simulation of two-phase energy and mass balance (STEAM), a compact simulator that models two-phase liquid cooling, focusing on the liquid-vapor phase change [16]. They model phase change from liquid to vapor, while we focus on phase change from solid to liquid. Tan et al. use computationally-intensive CFD simulations to analyze the PCM behavior, but do not consider real-life workloads [13]. Some other models use rather coarse-grained assumptions regarding the phase change duration and the PCM thermal properties [5]. Tilli et al. consider a more detailed PCM model, where they use a resistor-capacitor (RC) network and carry on latent heat energy calculations [6]. Their model assigns a *single RC* value for the whole PCM layer assuming homogeneous heat distribution across the PCM layer.

III. Modeling and Validation of Phase Change in Thermal Simulators

Having a detailed phase change model is essential for evaluating the benefits of various PCM material and volume choices as well as for analyzing the steady state and dynamic thermal behavior. To address this need, we propose a phase change model that can be integrated into compact thermal models and that is able to provide highly accurate temperature results within a short simulation time. This section explains our proposed modeling method in detail. It then continues with a discussion of how we implement the models used in prior work. Finally, we demonstrate the accuracy of our model by comparing it against COMSOL.

A. Proposed Modeling Approach

We leverage the compact modeling strategy for temperature modeling. In compact thermal simulators such as HotSpot [7], temperature is modeled based on an equivalent RC network, R and C representing the thermal resistance and capacitance, respectively. The temperatures of nodes are computed using this RC network and solving the corresponding differential equations. HotSpot models both lateral and vertical heat flow, as well as the chip package, including the heat spreader and the heat sink. HotSpot also allows the user to model basic 3D stacking by defining multiple layers of silicon, thermal interface material, or any other desired layer. Fine-grained simulation is carried out using the grid model, in which the floorplan is divided into smaller grids and temperatures are computed for each grid cell.

During phase change, PCM stores a large amount of energy at close-to-constant temperature, acting like a large thermal capacitor. The heat stored by PCM is called the *latent heat of*

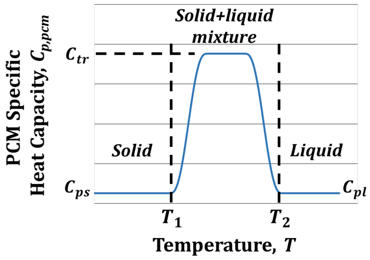


Fig. 1: Piecewise linear function for PCM specific heat capacity. Setting $c_{tr} \gg c_{ps}$ for the (T_1, T_2) interval models the phase change.

fusion and melting continues until PCM absorbs an amount of energy equal to its *latent heat of fusion*. Our goal, therefore, is to construct a model that can estimate the impact of phase change on temperature, as such a feature is not currently available in compact thermal simulators.

In this work, we focus on phase change from solid to liquid state and vice versa. We propose modeling phase change behavior using the *apparent heat capacity* method [12]. In this method, a nonlinear temperature-dependent specific heat capacity is assigned to the PCM layer as shown in Figure 1. The transition of the PCM from solid to liquid occurs over a temperature interval, where the specific heat capacity is very high compared to the material's heat capacity in the solid and liquid phases. Due to the high specific heat capacity, the rate of change of temperature is very low during phase transition. The integral of the heat capacity over the transition temperature range equals the *latent heat of fusion* for the PCM.

We implement this model in HotSpot as follows: We first define a layer of PCM material. The PCM layer is placed on top of the silicon layer and has the same layout as the silicon layer. Using the layer configuration files in HotSpot, we set the thermal conductivity and thickness of the selected PCM. We also modify HotSpot to define the melting point and latent heat of fusion of the PCM. Figure 2(a) shows the package layers for a chip with PCM. For thick PCMs, we divide the PCM layer into thinner layers in order to improve accuracy. It is important to use sufficiently thin layers of PCM for both accurate temperature calculation and modeling melting in the vertical direction. Figure 2(b) illustrates the grid cell structure for the silicon and thin PCM layers. The bottom layer is the silicon layer, which has the processing units where the heat is generated. On top of that is the PCM layer, which does not dissipate any power. Next, we assign each individual PCM grid cell a temperature-dependent specific heat capacity as in Equation 1:

$$C_{p,pcm}(T) = \begin{cases} c_{ps} & T < T_1 \\ c_{tr} & T_1 \leq T \leq T_2 \\ c_{pl} & T > T_2 \end{cases} \tag{1}$$

where c_{ps}, c_{pl}, and c_{tr} are the specific heat capacities of solid, liquid, and phase transition states, respectively. We use $c_{ps} = c_{pl}$ similar to prior work [17]. T_1 is the onset temperature and T_2 is the end temperature of the phase transition. In our experiments, we use a transition temperature interval of 3^oC [18], $c_{ps} = 1.57 \cdot 10^6 J/m^3K$, and $c_{tr} = 305 \cdot 10^6 J/m^3K$. Throughout the paper, we refer to the melting temperature as the center point of (T_1, T_2) interval. We implement the temperature-dependent heat capacity of Equation (1) using a smoothed piecewise linear function. At each time step, we update the specific heat capacity of each PCM grid cell depending on its temperature.

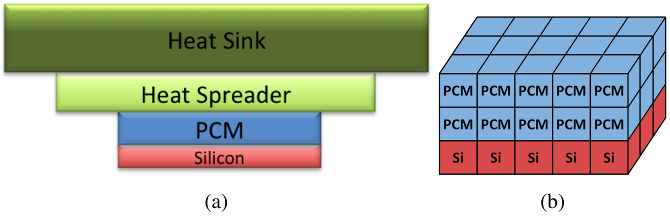


Fig. 2: (a) Package layers; (b) Silicon and PCM grid cells.

An important feature of our model is that it accounts for non-uniform melting of the PCM layer. In the case where there are idle and active cores, some portions of the PCM layer may melt earlier while other parts remain in the solid phase. Our model captures this behavior as we carry out phase change computations at a grid cell level. Throughout the paper, we use the notation *proposed model* to refer to our model.

B. Implementation of Phase Change Models in Prior Work

As briefly discussed in Section II, coarse-grained PCM models have been proposed and used in prior work [4], [6]. Raghavan et al. define an RC network for the silicon and PCM layers [4]. They use McPAT [19] to estimate the energy consumed by the cores. They then use these energy estimations to drive the RC model. However, the assumption that the energy dissipated by the cores is equal to the energy stored by the PCM is not highly accurate, because the energy stored in the PCM at a given time depends on the silicon temperature, PCM temperature, and PCM thermal resistivity. The package properties also affect the stored latent energy as some of the PCM energy is dynamically dissipated to the ambient environment. In addition, their model assumes perfect conductivity of the PCM combined with copper mesh, which may not be a feasible assumption within the chip package.

Tilli et al. propose a more detailed model where they use an RC network and a *latent heat energy* model [6]. We focus on this model for comparison purposes. In the Tilli et al. *latent heat energy* model, a lumped RC network is defined for the silicon layer. On the other hand, the PCM layer is treated as a single large cell where single R and C values are assigned to the whole layer assuming uniform heat distribution. During phase change, the PCM temperature is kept fixed at the melting temperature. To account for the melting duration, they compute the latent heat energy absorbed by the PCM using the heat transfer equation as follows:

$$\dot{U} = \sum_{k=1}^{N} \frac{T_k - T_{PCM}}{R_v} - \frac{T_{PCM} - T_{AMB}}{R_{PCM}} \tag{2}$$

where $\dot{U}$ is the rate of change of internal energy of the PCM, N is the number of silicon cells, T_k is the temperature of silicon cell k, T_{PCM} and T_{AMB} are the temperatures of the PCM layer and the ambient, respectively. R_v represents the contact resistance between the silicon and PCM layers in the vertical direction and R_{PCM} represents the thermal resistance of the PCM layer.

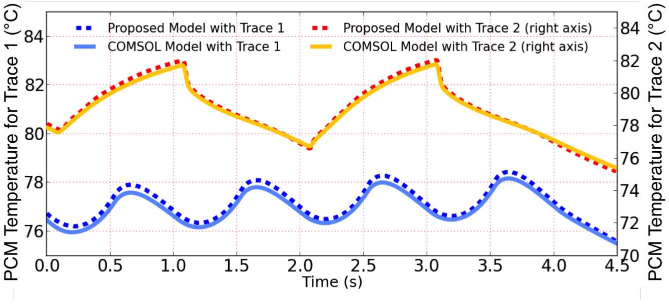


Fig. 3: Transient temperature comparison for two different power traces. Trace 1: square wave with 50% duty cycle; Trace 2: triangular wave with 1 sec period.

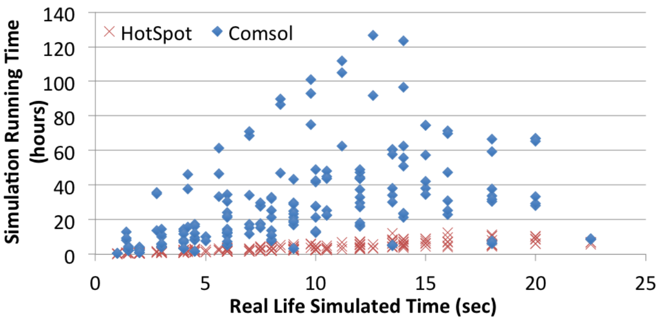


Fig. 4: Scatter plot comparing the solution time for COMSOL and HotSpot simulators.

We implement a model to mimic this *latent heat energy – single RC* model [6] in HotSpot. We define the PCM layer as a single large block for which HotSpot reports a single average layer temperature at each time step. In this model, the phase transition occurs at a layer granularity instead of a grid cell granularity. In other words, phase change starts when the layer temperature reaches the melting temperature and the temperature across the layer stays constant during phase change. The stored latent heat energy calculation happens at a layer level such that the aggregate energy stored in the PCM block is computed to decide if the PCM is fully melted or not.

We also implement a finer granularity version of this model, *latent heat energy – grid*, to show the impact of granularity. In this model, we define the PCM layer as an RC network and carry out temperature and latent heat energy calculations for each PCM grid cell. Note that *latent heat energy – grid* model is ***not*** included in prior work [6].

In summary, the key differences of our model compared to *latent heat energy – single RC* model of [6] are as follows: (1) we use the *apparent heat capacity* method to account for both temperature calculation and phase change duration, and (2) we carry out phase change computation at a grid cell level for the PCM layers.

C. Model Validation

This section provides a validation approach and demonstrates the accuracy of our phase change model by comparing it with COMSOL [8]. COMSOL models the chip package geometry as a set of 3D blocks stacked on each other, forming the layers of the package: silicon, PCM, heat spreader, and heat sink. The geometry is turned into a mesh composed of finely-sized tetrahedrals, comparable in size to the grid elements used in HotSpot. To model phase change behavior in the PCM layer, COMSOL uses the *apparent heat capacity* method [12]. COMSOL implements the two steps in the piecewise function of Equation (1) using a smoothed Heaviside function with continuous 2^{nd} derivative. This modeling method has been used in similar COMSOL simulations involving phase change behavior [17].

We carry out the validation experiments by simulating an AMD[2] Opteron 6172 processor (formerly codenamed "Magny-Cours"), using $8W$ and $2.63W$ for high and low power levels, respectively (see Section IV for details). Figure 3 compares the PCM layer transient temperature obtained by using our model against the COMSOL model. There are two example traces shown in the figure. Trace 1 uses a triangular wave with 1 second period for the power consumption signal and a 0.3 mm thick PCM with 77^oC melting point. Trace 2 uses a square wave with 50% duty cycle for the power signal and a 0.5 mm thick PCM with 80^oC melting point. Figure 3 shows that the temperature trace of our proposed model closely follows that of COMSOL. It should be noted that while the sophisticated COMSOL model is useful for validation, it runs far too slowly to evaluate the rapidly changing power traces we analyze in typical architectural simulations. Moreover, COMSOL requires several GB of storage even for a few seconds worth of real-life simulation; thus, it is not easily scalable to solve for longer traces. The scatter plot in Figure 4 compares the simulation running times of COMSOL and HotSpot for various benchmarks and simulation lengths. Our HotSpot implementation provides $37.5X$ maximum and $6.9X$ average simulation time savings in comparison to COMSOL. The running time difference between COMSOL and HotSpot is higher for benchmarks with rapid power variations.

We investigate the accuracy of our phase change model by running a large set of experiments using various power traces, PCM thicknesses, and melting points. In Table I, we report the maximum, mean, and standard deviation of error for a selected subset of these experiments. We present the temperature error across all units on both the silicon and PCM layers for our proposed model, as well as for the *latent heat energy – single RC* model [6], both compared against COMSOL. As highlighted in the table, the maximum temperature error is significantly larger for the *latent heat energy – single RC* model, reaching up to 9.18^oC. We also experimentally observe that the error of *latent heat energy – single RC* model is not sensitive to the transition temperature interval. On the other hand, our proposed model gives a maximum error of only 2.73^oC. The higher error occurs for benchmarks with abrupt power variations. We also compare our finer-granularity implementation, *latent heat energy – grid*, against COMSOL and observe a maximum error of 3.35^oC. Improving the *latent heat energy – single RC* model with *latent heat energy – grid* model reduces RMS error from 1.6^oC to 0.4^oC, and our proposed model reduces it further to 0.27^oC.

Figure 5 compares the thermal maps of the silicon layer for the two models at the time when 9.18^oC error is observed. The melting temperature is 85^oC. As seen from Figure 5(a),

[2] AMD, the AMD Arrow logo, AMD Opteron, and combinations thereof are trademarks of Advanced Micro Devices, Inc.

ERROR (°C)	PHASE CHANGE MODEL	Square Wave 25% Duty Cycle	Square Wave 50% Duty Cycle	Square Wave 75% Duty Cycle	Triangular Wave 1 sec Period	Triangular Wave 2 sec Period	bzip2	calculix	GemsFDTD	hmmer	lbm	leslie3d	gcc
MAX	Single RC[6]	7.59	7.78	9.18	7.52	6.96	3.45	7.33	5.95	6.33	4.21	4.71	5.26
	Proposed	1.23	1.11	1.08	0.9	0.9	1.93	1.99	2.22	2.73	0.76	1.33	2.59
MEAN	Single RC[6]	1.22	1.61	2.49	1.22	1.7	0.91	1.34	1.38	1.51	1.23	1.3	1.24
	Proposed	0.17	0.21	0.23	0.18	0.19	0.22	0.22	0.24	0.31	0.16	0.17	0.43
STD DEV	Single RC[6]	1.16	1.53	2.21	0.97	1.22	0.67	1.18	1.1	1.25	0.79	0.86	1.17
	Proposed	0.16	0.18	0.2	0.13	0.13	0.17	0.22	0.21	0.22	0.08	0.12	0.4

TABLE I: Maximum, mean, and standard deviation of error for the two melting models compared against COMSOL.

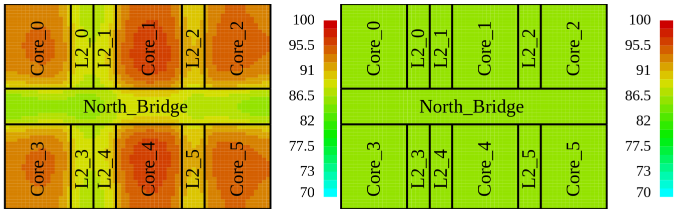
PCM portions that are layered on top of cores melt faster. The temperature starts rising when melting is complete, while the cooler areas such as the caches and the Northbridge controller are still in melting phase. On the other hand, in Figure 5(b), *single RC* model suggests that melting still continues all across the chip as the total PCM capacity is not fully exhausted, resulting in under-estimation of the core temperatures.

IV. Experimental Methodology

This section discusses our experimental methodology including the target system architecture, our workloads and the simulation framework. Our target system is based on the AMD Opteron 6172 processor. It is a large 12-core processor that consists of two 6-core Opteron processors connected by a HyperTransport pipe. It is manufactured using a 45 nm silicon on insulator process. We consider a single die with the 6-core processor and focus on the core and cache portions of the layout as shown in Figure 5. The core architectural parameters are taken from recent work [20].

Our simulation framework consists of four main parts: micro-architectural performance simulation, power simulation, temperature simulation, and bridge software with a database that decouples the performance-power simulation from thermal simulation. Figure 6 illustrates our simulation framework.

We use the Gem5 simulator [21] to collect detailed benchmark performance statistics. We fast-forward each benchmark for 2 billion instructions and collect the performance statistics in detailed mode every 1.5 million instructions with a total of 500 samples. Next, we feed those traces into the McPat 0.7 power modeling tool [19] to estimate the dynamic core power. We calibrate the runtime dynamic core power values using measurements collected on the AMD Opteron 6172 processor. The L2 cache power is calculated using CACTI 5.3 [22]. We scale the dynamic L2 power using the access rates. For our system, the average total power per active core is 7.5 W and the maximum cache power is 3.17 W. In HotSpot, we use the default package properties, except that we use 0.47 K/W for convection resistance and 45 J/K for convection capacitance.



(a) (b)

Fig. 5: Thermal map of the chip during melting for: (a) our proposed model; (b) *single RC* model.

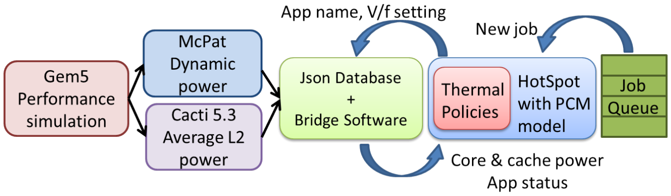


Fig. 6: Decoupled performance, power, and temperature simulation framework.

As the architectural level simulation is time consuming, we decouple the performance and power simulations from the temperature simulation, using an approach based on recent work [11]. We use the simplifying assumption that the behavior of individual applications is independent. This is reasonably accurate for systems with large private L2 caches as in our case. Based on our separability assumption, we generate a database of performance and power traces for each benchmark. The database acts as a lookup table where each cell corresponds to a 1.5 million instruction frame (for a given V/f setting). The cells store the core and cache power values and the running time corresponding to that frame. At runtime, HotSpot receives power data from the database by polling the bridge software at every sampling interval. We validate the separability assumption by running 60 test cases, comparing full instruction-level simulation and our method, and observe that the average IPC error is 1.16% and less than 2% for most test cases.

We run SPEC CPU2006 benchmarks as our workload. We select 13 benchmarks with varying performance characteristics. We assume the system has a job queue, where jobs out of the SPEC suite arrive following a Poisson distribution with exponential arrival times. We use arrival rates that correspond to 26% and 50% activity per core on average. The job queues are generated using Matlab and consist of job arrival times and the benchmark names. Our HotSpot implementation interfaces with the job queue file.

V. Results

In this section, we first show the impact of PCM properties on the temperature profile of a processor. We then compare our proposed model against the *single RC* model and discuss the significance of modeling accuracy. Finally, we evaluate the behavior of a thermally-aware throttling policy using PCM. We experiment with various PCM configurations and melting points, and present a subset of these cases to help visualize our points in the most clear way.

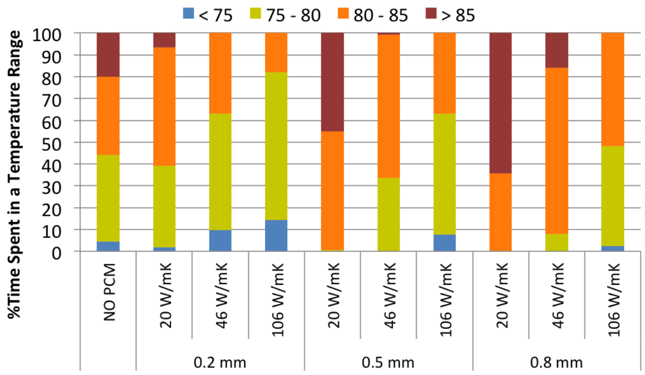


Fig. 7: Percent of time spent within temperature ranges for 9 PCM configurations.

A. Design Space Exploration of PCM Properties

PCM materials vary in their latent heat of fusion values and melting temperatures. The larger the latent heat of fusion, the more heat we can store during melting. Typical values for melting temperature are 30 to 70^oC [23]; however, we explore a wider range of temperatures. It is possible to adapt the power density of cores and PCM parameters to meet melting temperature constraints if needed. In many applications, it is desirable to have the melting temperature close to, but below the maximum allowed chip temperature. In this way, we can make the most of the latent heat storage ability of the PCM. The melting point should not be too low because once the PCM is fully melted, it will need to be cooled down to that temperature again to start freezing. Thus, low PCM melting temperatures either keep the PCM melted in the common case (rendering it ineffective), or may cause frequent throttling if the system is desired to operate close to the melting temperature.

The thermal conductivity of the PCM is also an important parameter. It is desirable to have high conductivity to help homogeneous melting/freezing as well as to avoid overheating. Higher conductivity PCM results in lower maximum temperature as it provides a smaller equivalent thermal resistance between the silicon layer and the heat sink. Cerrobend (19 W/mK) and gallium (33.7 W/mK) are examples of higher conductivity PCM [23]. Most other PCMs, such as paraffin, have very low conductivity. Conductivity enhancement techniques can overcome this challenge, such as embedding the PCM into a metal matrix [24], [25].

The amount of PCM also impacts temperature profiles strongly. While a thick PCM can maximize the amount of heat absorbed and delay entry into fully melted (liquid) state, it can also interfere with the effectiveness of the heat sink after melting is complete. This is because the relatively lower conductivity of the PCM can reduce the efficiency of heat transfer to the high-conductivity heat sink.

In our design-space exploration, we mainly focus on the conductivity, melting point, and the thickness of the PCM as design parameters. We assume the use of highly thermally-conductive copper-PCM matrix [25] with various PCM fractions, and explore the impact of PCM for 0.2-0.8 mm thickness and 20-106 W/mK conductivity (corresponding to PCM fractions 100%-70%). We set the melting temperature as 80^oC and the total simulation time as 10 seconds. Figure 7 shows the percentage of time spent within different temperature ranges for 9 different PCM configurations as well as for no-PCM case. We see that the PCM properties have a significant effect on the temperature profile of the processor. For higher PCM conductivity, cores spent less time in the high temperature range. This impact of conductivity becomes even more apparent for the 0.8 mm PCM. While the amount of time spent in the highest temperature range is 65% for 20 W/mK PCM, it decreases to 15% and 0% for 46 W/mK and 106 W/mK PCMs, respectively. In terms of the maximum and average temperatures, we observe up to 20.1^oC and 11.7^oC difference, respectively, among the 9 PCM configurations. Another interesting result is that choosing the wrong PCM may result in higher temperatures than having no PCM at all. For example, for the system with no PCM, the temperature exceeds 85^oC 20% of the time; while for the 0.5mm, 20 W/mK PCM, it rises to 45% as the poor conductivity of the PCM interferes with the effectiveness of the heat sink.

In general, having the highest conductivity PCM available is preferred for all cases. However, the cost and availability of a high conductivity material becomes a trade-off. On the other hand, choosing the best thickness is not trivial. Without any control mechanism, higher thickness will result in higher peak temperature once the melting finishes. However, applying PCM-aware control mechanisms (e.g., [6]) may make the thicker PCM more attractive.

B. Impact of Modeling Accuracy on System Design

In this section, we compare our model against the *single RC* model [6] and show the benefits of our detailed phase change modeling empirically. To show the differences in the behavior of the two models, we use a dynamic workload queue as discussed in Section IV. For the first set of experiments, when a new job arrives in the queue, we initiate the same job on all 6 cores. We present two main differences in the thermal behavior of the two models.

Implication 1: Time spent before starting melting. Defining the PCM layer as a single large cell makes it harder to start melting. Figure 8 illustrates this point, where the PCM layer temperatures and the percentage of PCM melted are plotted over time. The melting point is set to 85^oC. Note that for the *single RC* model, the reported PCM temperature corresponds to the whole layer, while for the proposed model, it corresponds to a specific portion of PCM that lies on top of Core_0. Our proposed model suggests that PCM starts melting at around 0.5 seconds and % of melted PCM increases as shown by the blue curve; while for the *single RC* model, % of melted PCM stays at zero until 1.5 seconds. There are two main reasons for that: (1) For the *single RC* model, the average temperature across the PCM has to reach the melting temperature to start melting. However, the impact of the hotter core temperatures is averaged out by the colder units on the chip, resulting in a lower reported PCM temperature. (2) In reality, the PCM grid cells that are closer to the silicon layer in the z-direction reach melting before the ones that are further away from the silicon layer. The proposed model captures this effect, which is why the % of melted PCM curve starts rising before the average PCM temperature reaches 85^oC.

Implication 2: Constant temperature assumption during melting. For the *single RC* model, once the melting starts,

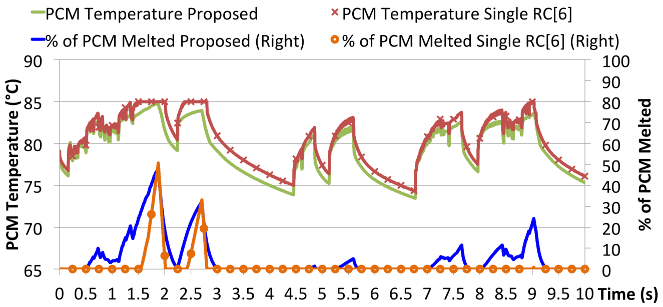


Fig. 8: Implication 1: *single RC* assumption results in longer time before starting melting.

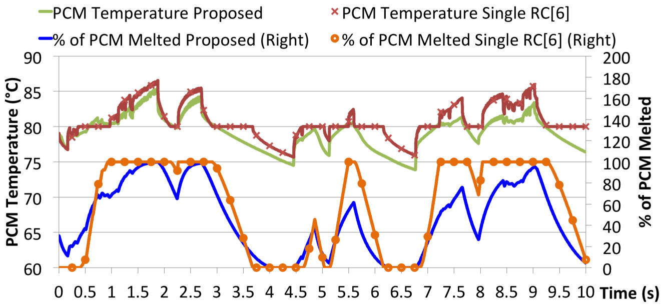


Fig. 9: Implication 2: *single RC* assumption results in constant temperature until 100% of the PCM is melted.

the PCM temperature rises (falls) above (below) the melting temperature only when the % of melted PCM reaches 100% (0%). However, in reality, different portions of the processor consume the PCM latent heat capacity at different rates. Thus, for example, even when 70% of the PCM is melted, the temperature of some PCM portions can be higher than the melting temperature. Figure 9 shows this empirically: melting point is set to 80^oC, the PCM temperature line of the *single RC* model stays constant at 80^oC until the % of melted PCM line reaches 100% at around 1 second. On the other hand, the temperature line of the proposed model suggests that the portion corresponding to the Core_0 area fully melts and the temperature of that area starts rising at around 0.5 second.

The effect of these implications becomes even more significant for the case where the heat distribution across the chip is highly heterogeneous. In order to illustrate this point, we simulate a scenario in which the incoming job of the same queue is initiated on only one of the cores of the processor (Core_0), while the other cores stay idle. Melting point is set as 75^oC. Figures 10(a) and (b) show the resulting temperatures of the PCM and Core_0, respectively. We see a significant difference in the temperature traces reported by the two models. The *single RC* model suggests that the PCM temperature never reaches melting point throughout the simulation. As a result, the core temperature keeps rising for the *single RC* model. On the other hand, our model shows that the PCM portion that corresponds to the Core_0 area starts melting, as indicated by the blue line in Figure 10(a). The impact of this difference in the estimated PCM temperature results in an over-estimation of Core_0 temperature for the *single RC* model, as shown in Figure 10(b). Our model detects that melting has started on the Core_0 portion of the PCM, thus, the rise in core temperature slows down owing to the heat absorption during phase change.

TABLE II: Number of throttling instances for each core using the two models.

Number of Active Cores	**Proposed Model**						**Single RC Model [6]**					
	Core Number						**Core Number**					
	0	**1**	**2**	**3**	**4**	**5**	**0**	**1**	**2**	**3**	**4**	**5**
1	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	3	3	0	0	0	0
3	0	4	0	0	0	0	19	77	30	0	0	0
4	6	18	2	2	0	0	71	141	49	46	0	0
5	14	48	9	12	23	0	94	186	64	84	129	0
6	22	84	30	22	84	30	10	28	13	10	28	34

The results of Figures 10(a) and (b) are especially important when considering PCM applications such as *computational sprinting*. In *computational sprinting*, there are two operating modes: *sustained* mode in which a single core is activated, and a *sprinting* mode in which all cores of the processor are active, exceeding the TDP of the chip. The same per core active power is used for both *sustained* and *sprinting* modes. For the *sustained* mode operation, it is assumed that melting does not occur at all. As we have discussed above, in fact, localized melting can occur even during sustained operation, and the particulars of the sustained execution will impact the total latent heat capacity available for sprinting.

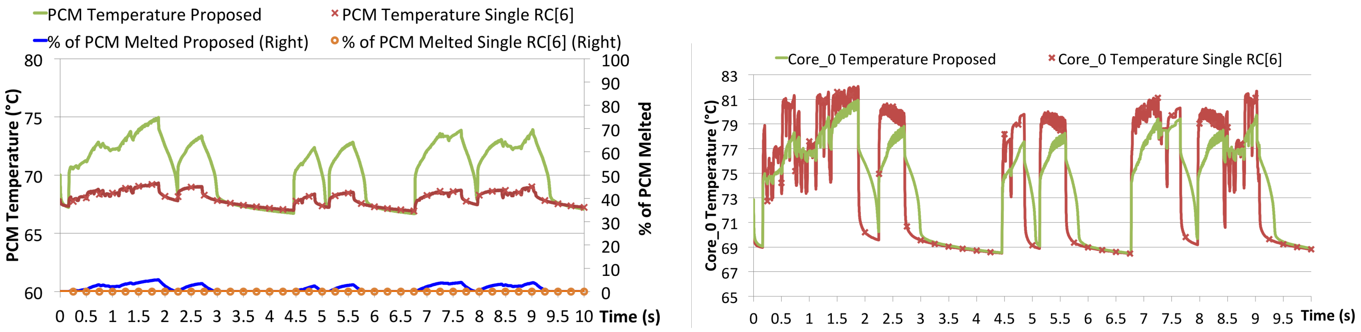
C. Evaluation of Runtime Management Policies

In this section, we evaluate the impact of modeling on the runtime management policy decisions. The previous experiments do not use any thermal management policies, in order to show the sole effect of modeling on the temperature. A more realistic scenario is where PCM is used in cooperation with the dynamic thermal management strategies. In this work, we show that a better modeling approach changes the design time evaluation of those strategies.

For this purpose, we implement a temperature-aware throttling policy and evaluate the behavior of the policy using the two phase change models. In this policy, if the core temperature exceeds a predefined upper threshold, it is put to idle for a fixed amount of throttling time. We use 80^oC as the temperature threshold and 10 ms of throttling time, which is used in current systems. At the end of the throttling time, the policy checks if the temperature has fallen below the threshold value and if not, triggers the mechanism again. We use the low utilization queue described in Section IV and run the simulations using different number of active cores changing from 1 to 6. We record the number of instances each core is throttled for the two models.

Table II shows that when only a few cores are active, the *single RC* model over-estimates the core temperatures, leading to higher number of estimated throttling. This is in line with Implication 1, as well as the behavior explained in Figure 10. For lower utilizations, the PCM layer temperature reported by the *single RC* model does not reach melting temperature even though the temperatures of active cores are rising. Thus, in comparison to the proposed model, at lower utilizations, it over-estimates the number of throttling instances by 31x, 10x, and 5x for 3, 4, and 5 active cores, respectively. Our model, on the other hand, captures the localized melting behavior and reflects the benefit of phase change on the core temperature.

As the number of active cores increases and the melting



(a)

(b)

Fig. 10: Comparison of PCM and core temperatures for the two models when only Core_0 is activated. (a) Lower PCM temperature is observed for *single RC* model. (b) Lower PCM temperature suggests no melting, leading to over-estimation of Core_0 temperature.

occurs, the *single RC* model starts to under-estimate the core temperatures and the number of throttling instances. This is a result of Implication 2; if there is melting, the PCM temperature (thus, the core temperature) in *single RC* model will stay constant for a longer time in comparison to our proposed model. An example case is with 6 active cores, for which both models show that PCM goes through melting within the simulated time. For that case, *single RC* model under-estimates the number of throttling instances by 2.6x.

VI. Conclusion

This research presents new techniques for modeling the interaction of phase change materials on or near the CPU chip. By modeling the material at a fine granularity, it enables the model to exhibit complex behaviors that prior models have missed. This greatly increases the accuracy of the model, enabling this technology to move forward with better assessments of the opportunities of phase change material and more accurate tuning of policies and mechanisms.

References

[1] A. K. Coskun, D. Atienza, T. S. Rosing, T. Brunschwiler, and B. Michel, “Energy-efficient variable-flow liquid cooling in 3D stacked architectures,” in *DATE*, pp. 111–116, 2010.

[2] H. Xie, A. Ali, and R. Bhatia, “The use of heat pipes in personal computers,” in *ITHERM*, pp. 442–448, 1998.

[3] S. Biswas, M. Tiwari, T. Sherwood, L. Theogarajan, and F. Chong, “Fighting fire with fire: Modeling the datacenter-scale effects of targeted superlattice thermal management,” in *ISCA*, pp. 331–340, 2011.

[4] A. Raghavan, Y. Luo, A. Chandawalla, M. Papaefthymiou, K. P. Pipe, T. F. Wenisch, and M. M. K. Martin, “Computational sprinting,” in *HPCA*, pp. 1–12, 2012.

[5] A. Raghavan, L. Emurian, L. Shao, M. Papaefthymiou, K. P. Pipe, T. F. Wenisch, and M. M. Martin, “Computational sprinting on a hardware/software testbed,” in *ASPLOS*, pp. 155–166, 2013.

[6] A. Tilli, A. Bartolini, M. Cacciari, and L. Benini, “Don’t burn your mobile!: safe computational re-sprinting via model predictive control,” in *CODES+ISSS*, pp. 373–382, 2012.

[7] K. Skadron, M. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan, “Temperature-aware microarchitecture,” in *International Symposium on Computer Architecture, ISCA.*, pp. 2–13, 2003.

[8] “Comsol Multiphysics.” http://www.comsol.com.

[9] M. Bao, A. Andrei, P. Eles, and Z. Peng, “On-line thermal aware dynamic voltage scaling for energy optimization with frequency/temperature dependency consideration,” in *Design Automation Conference (DAC)*, pp. 490–495, 2009.

[10] A. Bartolini, M. Cacciari, A. Tilli, and L. Benini, “Thermal and energy management of high-performance multicores: Distributed and self-calibrating model-predictive controller,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 1, pp. 170–183, 2013.

[11] A. K. Coskun *et al.*, “Evaluating the impact of job scheduling and power management on processor lifetime for chip multiprocessors,” in *SIGMETRICS*, pp. 169–180, 2009.

[12] E. Alawadhi and C. H. Amon, “Pcm thermal control unit for portable electronic devices: experimental and numerical studies,” *IEEE Transactions on Components and Packaging Technologies*, vol. 26, pp. 116–125, March 2003.

[13] F. Tan and S. C. Fok, “Thermal management of mobile phone using phase change material,” in *Electronics Packaging Technology Conference*, pp. 836–842, 2007.

[14] D. won Yoo and Y. Joshi, “Energy efficient thermal management of electronic components using solid-liquid phase change materials,” *IEEE Transactions on Device and Materials Reliability*, vol. 4, no. 4, pp. 641–649, 2004.

[15] A. Stupar, U. Drofenik, and J. Kolar, “Application of phase change materials for low duty cycle high peak load power supplies,” in *International Conference on Integrated Power Electronics Systems (CIPS)*, pp. 1–11, 2010.

[16] A. Sridhar, Y. Madhour, D. Atienza, T. Brunschwiler, and J. Thome, “Steam: A fast compact thermal model for two-phase cooling of integrated circuits,” in *ICCAD*, pp. 256–263, Nov 2013.

[17] W. Ogoh and D. Groulx, “Effects of the heat transfer fluid velocity on the storage characteristics of a cylindrical latent heat energy storage system: a numerical study,” *Heat and Mass Transfer*, vol. 48, no. 3, pp. 439–449, 2012.

[18] V. S. S. Srinivas and G. K. Ananthasuresh, “Analysis and topology optimization of heat sinks with a phase-change material on COMSOL multiphysics platform,” in *COMSOL Users Conference*, pp. 1–7, 2006.

[19] S. Li *et al.*, “McPAT: An integrated power, area, and timing modeling framework for multicore and manycore architectures,” in *MICRO*, pp. 469–480, 2009.

[20] P. Conway, N. Kalyanasundharam, G. Donley, K. Lepak, and B. Hughes, “Blade computing with the AMD Opteron Processor (magny-cours).” http://www.hotchips.org/wp-content/uploads/hc_archives/hc21/2_mon/HC21.24.100.ServerSystemsI-Epub/HC21.24.110.Conway-AMD-Magny-Cours.pdf, Aug. 2009.

[21] N. L. Binkert *et al.*, “The M5 simulator: Modeling networked systems,” *IEEE Micro*, vol. 26, pp. 52–60, July 2006.

[22] S. Thoziyoor *et al.*, “CACTI 5.1,” tech. rep., April 2008.

[23] D. Hale *et al.*, “Phase change materials handbook,” 1971.

[24] A. Mills, M. Farid, J. Selman, and S. Al-Hallaj, “Thermal conductivity enhancement of phase change materials using a graphite matrix,” *Applied Thermal Engineering*, vol. 26, no. 1415, pp. 1652 – 1661, 2006.

[25] S. Lingamneni, M. Asheghi, and K. Goodson, “A parametric study of microporous metal matrix-phase change material composite heat spreaders for transient thermal applications,” in *ITHERM*, May 2014.