Variation-aware Thermal Characterization and Management of Multi-core Architectures

Eren Kursun, Chen-Yong Cher

IBM Thomas J. Watson Research Center
Yorktown Heights

***Abstract* –The accuracy and efficiency of dynamic power and thermal management are both affected by the increased levels of on-chip variation, mainly because dynamic thermal management schemes are oblivious to the variation characteristics of the underlying hardware. We propose a technique that utilizes the existing on-chip sensor infrastructure to improve the inherent thermal imbalances among different cores in a multi-core architecture. Thermal sensor readings are compiled to generate an on-chip variation map, which is provided to the system power/thermal management to effectively manage the existing on-chip variation. Experimental analysis based on live measurements on a special test-chip shows reduced on-chip heating with no performance loss, which improves the power/thermal efficiency of the chip at no cost.**

I. INTRODUCTION

In recent years, increased on-chip variation was observed at various granularities: Individual cores, functional units and even macros on the same die differ in terms of performance, peak clock frequency, power and thermal profile. Variation in parameters such as V_{th}, L_{eff}, t_{ox} and pattern density has been increasing steadily [3]. Timing and functional implications of such on-chip variation are subject to many research studies [2], [5], [11], [16], [20].

However, research on power/thermal aspects of on-chip variation is limited; mostly due to the belief that such variation would be minimal at core or functional unit level. Our experimental analysis based on thermal imaging of real chips indicates that such expectations are not well grounded; and that, significant thermal variation can be observed even at core/block level in the current process technologies.

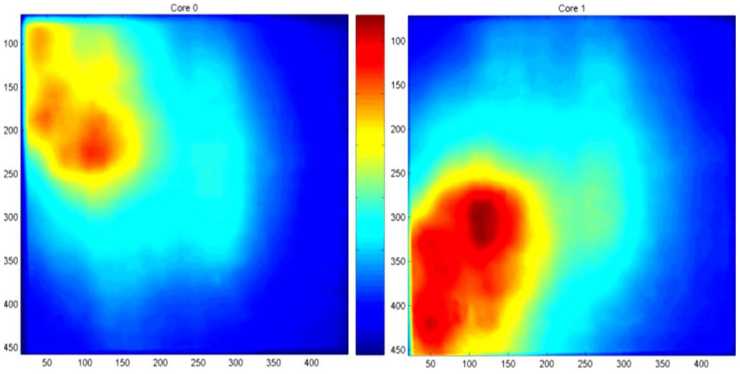
Although, a number of dynamic thermal management techniques have been proposed toward reducing the on-chip heating, these techniques are mostly oblivious to the underlying variation. Unlike the heating behavior on ideal chips with no variation; such heating is more difficult to plan for and evaluate during the initial design and planning stages.

On-chip variation interferes with the accuracy of thermal management schemes, such as task profiling, since the existing thermal management techniques do not differentiate whether the heating is caused by task characteristics or by the underlying hardware tendencies. Hence, traditional DTMs may even incorrectly profile hot jobs as cold, or cold jobs as hot. In such cases, unplanned heating behavior along with inaccurate thermal management can cause more frequent thermal emergency responses; causing performance degradation.

In this study we investigate the effects of variability on the existing process technologies by real-life analysis of a special process variation test-chip. This enables the following observations and contributions:

- Both infra-red thermal imaging and sensor measurements consistently confirm and quantify the core-to-core and unit-to-unit thermal variation on a real test chip.
- We study the correlation between workload characteristics and chip heating behavior using performance counters and thermal sensors. To the best of our knowledge, this is the first high fidelity study of temperature and performance counter correlation in both single-core and multi-core processors.
- We propose a novel variation assessment technique, which leverages a sensor based characterization stage to generate a high-level chip variation map. This map is then used to improve the efficiency of dynamic power/thermal management of the multi-core architecture.

Since the variation profile is unique to each chip, an effective solution needs to tune into the individual characteristics of the underlying hardware. We show that run-time assessment using on-chip sensor infrastructure can substitute for manufacturers' guidelines. This enables dynamic power/thermal management to adapt to the underlying chip characteristics with no additional performance degradation. Experimental analysis on a test chip shows that the efficiency of thermal management techniques such as activity migration and thermal-aware scheduling can be improved through such variation-awareness. As a result, power and thermal efficiency of the chip is improved at no cost. (Power improvement is mostly the result of the temperature-leakage dependency, which becomes prominent at high temperatures). Please note that the main focus of this study is the power/thermal variation on the chip. Functional and timing implications are beyond the scope.



(a) *daxpy* on core 0 (b) *daxpy* on core 1

Figure 1. Measured thermal variation on special test chip

Current technology trends indicate increases in all major

978-1-4244-2658-4/08/$25.00 ©2008 IEEE

components of on-chip variability including: L_{eff}, V_{th}, t_{ox} imperfections, supply grid/V_{dd}, and W,T,H variations [2]. The underlying hardware imperfections exhibit themselves as differences in: timing characteristics, leakage power dissipation and temperatures. In addition, imperfections in the packaging and cooling system can contribute to the on-chip differences. As a result, the core-to-core variation in a multi-core architecture can be quite significant [17]. Figure 1 illustrates the temperature difference between the two cores running the same benchmark, daxpy, on the same test-chip. (Further details of the experiments/setup are discussed in Section III)

In general, the functional/timing problems due to core-to-core variation can be addressed by setting the clock frequency (and supply voltage) per core, or according to the slowest core on the chip [22]. However, power/thermal differences among cores are largely unaddressed. Note that, on-chip sensors do not guarantee the efficiency of dynamic power/thermal management under process variation, which affects the functionality of the DPTM schemes in various ways:

- Underlying variation distorts the thermal task profiling: Hence, power/thermal management schemes are unable to identify whether the heating is caused by the inherent tendency of the underlying hardware unit or the characteristics of the task running on it. Cool jobs may be incorrectly profiled as hot and hot jobs as cold. As a result, assuming that the hardware is uniform and using incorrect profiling data power/thermal management will not be effective in assigning the tasks.
- The cores with higher leakage power dissipation have an inherent tendency to heat up more frequently than the others. This is valid even when all the cores start at the same ambient temperature. Since traditional dynamic thermal management schemes do not take this into account, the resulting heating need to be addressed by throttling the resources. As a result, number and frequency of such unplanned hotspots can cause performance degradation.
- Even a few degrees of temperature increase translates to higher leakage power at normal server operating temperatures, which causes increased power/cooling/maintenance costs for the data center. Power/cooling costs constitute about 40% of the overall data center cost, and as high as 60% of the running cost, according to the data from US Department of Energy [21]. Due to the exponential dependency between on-chip temperatures and leakage power, even a few degrees of temperature reduction is likely to translate to observable reduction in the data center running cost.

II. VARIATION-AWARE THERMAL MANAGEMENT

Figure 2 shows the overview of the proposed technique, which starts with a high-level characterization of the chip by collecting sensor data during isolated runs over individual cores and architectural units. While it is not possible to completely isolate the computation to a single architectural block, special benchmarks that stress the individual architectural units are used for block characterization.

After the data is collected, the differences among blocks are analyzed to generate a variation map. The deviation of the block temperature from chip average and from blocks with identical functionality is calculated for different starting temperatures and workloads. The multiple data points indicate the leakage differences of the core/block relative the other cores/blocks. Each temperature reading is compared to the corresponding hardware counter, which indicates whether the heating is inherent to the block or it is caused by the utilization. The block temperatures are ranked in terms of criticality, such that high temperature blocks are identified properly. These three components (temperature deviation, activity counter comparison and block criticality) are then compiled to represent the variation coefficient of each block. Similarly, a coefficient is assigned per core.

During the profiling phase, it is important to collect thermal sensor readings in isolation (for individual cores/functional units) over separate runs to isolate the characteristics and filter out the thermal spills from neighboring blocks. This is a key requirement to accurately profile the chip. While various on-chip sensors such as temperature sensors and critical path monitors can be used for this profiling phase, we focus on thermal sensors due to wider availability. The variation map is maintained by the system software specifically by power/thermal management schemes. The table can also be maintained in hardware as long as the information is made transparent to the system software and dynamic power/thermal management schemes.

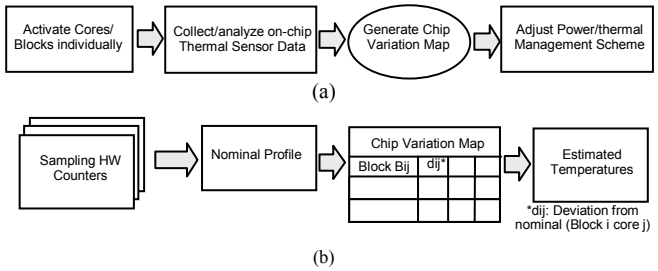


Figure 2. (a) Generating the variation map for individual cores; (b) Using the variation map for accurate temperature estimation at run-time

The variability information can be acquired in a number of ways including performance measurements per core, temperature measurements, or manufacturer guidelines. Even though manufacturer guidelines could provide the simplest solution, currently we are not aware of any manufacturer providing such information.

Furthermore, the variability profile is different for each individual chip. Hence, it is an effective of incorporating a universal and self adjusting scheme, which is capable of addressing all possible cases. As a result, we focus on assessment of variation by the on-chip sensors (chosen for the wide-spread availability) as well as utilizing this information at the system software level. Hence the proposed technique can easily be used by various software thermal management schemes running on existing multi-core architectures with on-chip thermal sensors as well.

Variation-awareness can be incorporated into a wide range of thermal management schemes. We illustrate the potential benefits through two DTM schemes: (1) *Core hop-*

ping, balances the on-chip profile by moving the computation from hotter cores to cooler ones (2) *Thermal-aware task scheduling,* which uses thermal task profiling to effectively assign tasks on the chip.

Variation-awareness enables core hopping and task assignment schemes to compensate for the inherent differences between the cores/units, by utilizing the cooler cores more heavily than the hotter counterparts. Similarly, thermal-aware task scheduling uses the on-chip variation map to accurately profile the threads and tasks to effectively manage the hardware resources to meet software requirements.

III. METHODOLOGY

Experimental data was collected on a specially selected test chip at 1.5 GHz, and 1.05 V running Bare Metal Linux [23]. Please note that it is possible to run at higher supply voltages and clock frequencies for which the thermal benefits of the proposed schemes are more pronounced. However, due to the thermal imaging framework limitations lower operating voltages were chosen for the runs. Hence, the experimental results should be interpreted as conservative indicators of potentially higher improvement in terms of temperatures.

On-chip temperature sensor data was sampled by the BML at each scheduling tick (10ms). SPEC2006 benchmark set is used for the experimental analysis including: integer *perlbench, mcf, hmmer, libquantum* and floating point benchmarks: *milc, gromacs*, *namd*, *daxpy*, and *lucas*. Figure 3 shows the experimental setup used for the thermal analysis: The on-chip sensors are calibrated by comparing the infra-red images [9]. This kind of calibration is not required for the proposed scheme in general, but it was employed for experimental analysis purposes. (Most manufacturers pre-calibrate the sensors before the chips are shipped.)

The setup in Figure 3 was employed to generate the thermal images presented in the experimental result section. The imaging setup required replacing the traditional cooling solution with a transparent liquid, to enable the infrared camera take real-life measurements of the underlying chip. The heat sink was replaced during the calibration and imaging phase with a liquid heat sink to enable the infra-red imaging analysis. Later, the heat sink was reattached and the runs were repeated to ensure consistency.

It is also important to note that variation characteristics are unique to the individual chips and the presented profile is not likely to represent other chips, hence a static solution can not be applicable. However, the proposed technique can be applied to any multi-core architecture; it's capable of handling the unique characteristics of any chip with variation. Table 1 lists further info on the architecture used for the experimental analysis:

Table 1. Processor parameters

1.5GHz cores	1.05V
1.44MB L2	2GB Memory

IV. EXPERIMENTAL ANALYSIS AND RESULTS

In this section we present measured data from live measurements on a test chip as discussed earlier in Section III. We start with high-level chip characterization, which is used to generate the variation map. Then, we present improved core hopping and task scheduling enabled by this map. Please note that the measured temperatures are dependent on the characteristics of the experimental setup and corresponding cooling solution; these absolute temperature values should not be interpreted as typical operating temperatures.

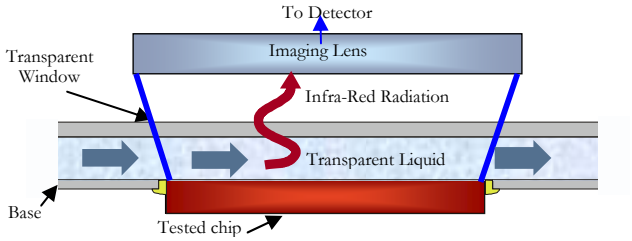


Figure 3. Experimental setup for real-time thermal imaging

A. Core-to-Core and Within-Core Thermal Variation

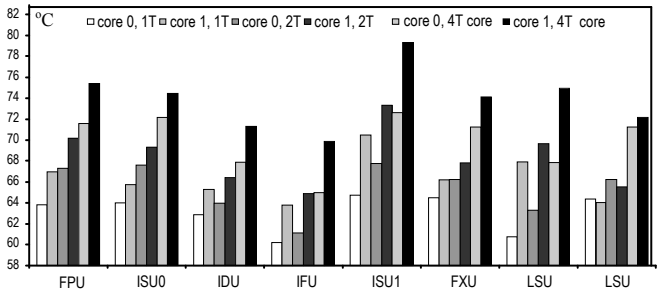


Figure 4a. Peak temperature values for blocks in core 0 and core 1 for SPEC2k6 (FPU: Floating Point Unit, ISU: Instruction Scheduler, IFU: Instruction fetch Unit, FXU: Fixed Point Functional Unit, IDU: Instruction Dispatch Unit, LSU: Load Store Unit)

Figure 4a displays the measured variation between the two cores on the test chip, running 1-4 threads in single-threaded and simultaneous multi-threading (SMT) modes. Core 1 is almost always hotter than core 0 (except for the LSU1, which we will discuss later). The temperature difference between the cores is as high as 6°C for ISU1 and 7.5°C for LSU0 during the experiment runs. ISU1 was the hotspot over all the runs, with peak temperature around 80°C (when both cores are active in SMT2 mode). Notice that the presented results include isolated core runs used for the variation map as well as multi-threaded runs for general characterization. Figure 4a also reveals that thermal profiles within the two cores are not identical. For instance, load store unit, LSU0, is one of the hotspots on core 1; whereas it is relatively cooler on core 0. On the other hand LSU1 has the opposite thermal profile, an elevated temperature for core 0 than core 1. Similar thermal differences are observable on other blocks.

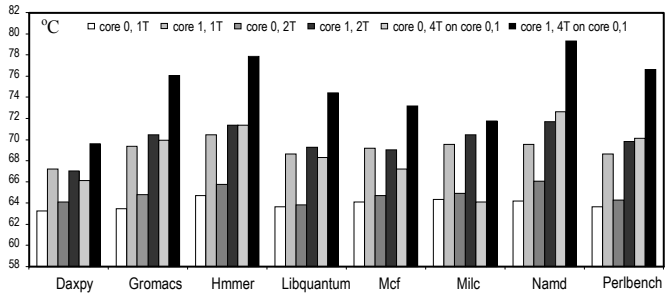


Figure 4b. Peak core temperatures for different SPEC2k6 benchmarks

Figure 4b shows that the temperature differences are consistent over all the experimented benchmarks from SPEC2k6 suite. On average, the peak core temperature is 4°C higher on core 1. Simultaneous multi-threading causes further increase in the on-chip temperatures. The existing hotspots with high utilization become even more prominent under SMT (such as ISU1, FPU etc), due to the differences in utilization.

B. Performance Counter-Thermal Sensor Correlation

The correlation between the performance counters and the temperature profile is analyzed in this section: Figure 4a indicates that the temperatures can be attributed to the increase in core utilization. The only exception is daxpy, which has the highest IPC but lowest temperature among the benchmarks.

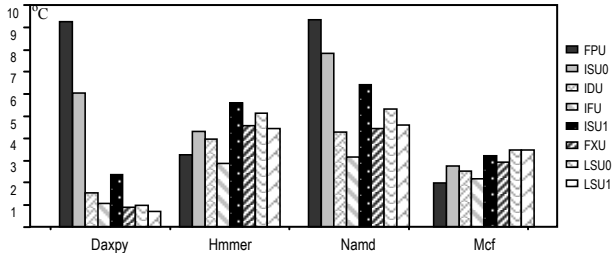


Figure 5. Increase in the block temperatures for benchmarks in SMT2 mode (y-axis displays the temperature increase in °C)

Figure 5 shows the temperature increase per unit in SMT2 mode compared to Linux idle loop, indicating the correlation between unit temperatures and workload demands. From Figure 5, namd and daxpy heat the FPU because of the high utilization of the unit. Both hmmer and mcf do not utilize the FPU, yet FPU temperatures are different because of the overall heating. Therefore, we conclude that instructions per cycle and unit activities, together with our variability assessment scheme should be taken into account for temperature estimations.

Figure 6a and 6b demonstrates the correlation between lucas and milc, as well as their phase behavior: The top part of the figure shows the thermal sensor readings; the bottom is the performance counters normalized to the number of cycles. Notice that the activity counter peaks are followed by the thermal peaks with a delay; which is due to the thermal time constant, same effect is observable for cooling period. The thermal time constant for this case is in the order of 50 msecs.

Another point to note is the difference between milc and lucas temperature profiles. The length and intensity of high-activity period in lucas cause the temperatures to rise considerably in the FPU, whereas the rapid switches between high and low activity modes do not translate to immediate temperature difference in milc. This clearly illustrates the low-pass filter character of temperature on the underlying hardware activity. Figure 6c shows the correlation between the hardware counters and on-chip temperatures for FPU. The highlighter sampling interval of 50 msec at the bottom left gives the one-to-one correlation between the counters and on-chip temperature readings.

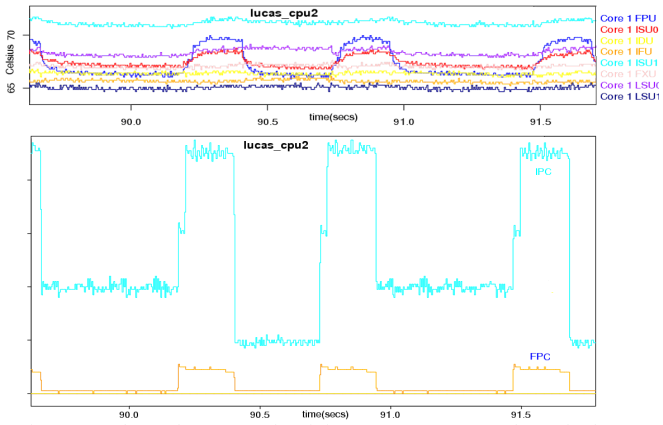


Figure 6a. Thermal sensor and activity counters (FPC) and IPC for lucas

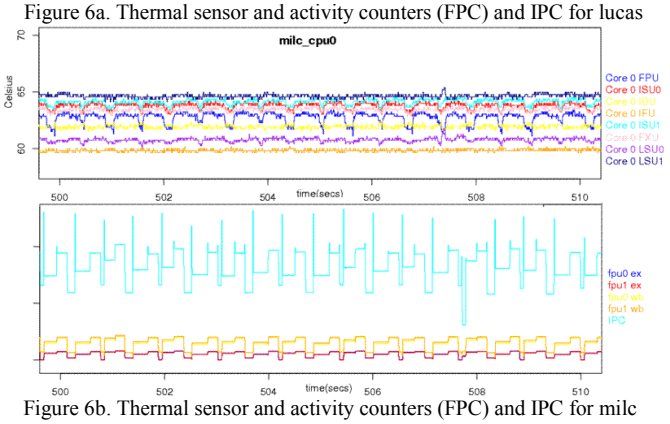


Figure 6b. Thermal sensor and activity counters (FPC) and IPC for milc

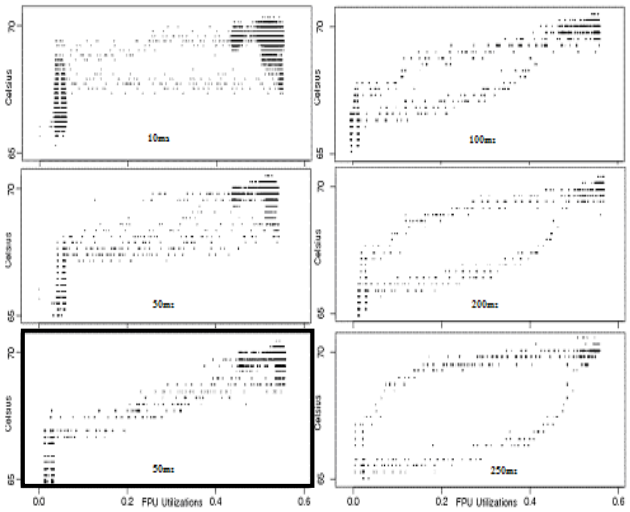


Figure 6c. Hardware counter-temperature correlation for FPU for different sampling intervals for lucas: y-axis: Celsius, x-axis: FPU utilization for 10-250 msec range.

C. Variation-aware Core Hopping

We experimented with pre-emptive activity migration to demonstrate the potential improvement in variability-awareness. Even though reactive activity migration is equally applicable, our goal is to avoid the heating by distributing the computation between two cores effectively. The core hopping interval is adjusted to adapt to the variation between the intrinsic thermal profiles of the two cores.

Figure 7a and 7b illustrate the effects of core hopping on peak block temperatures within the core over various

benchmarks. Equal interval activity migration is compared to variability-aware activity migration (at the bottom) for namd in Figure 8. By choosing the core hopping intervals to be shorter on the hotter core 1, the corresponding hotspots are reduced (100ms/Core 0 - 10ms/Core 1), whereas the core 0 does not heat up significantly due to the higher utilization.

The effects of symmetrical and asymmetrical intervals are displayed for core 1 running hmmer, as shown in Figure 9 shows the resulting thermal image of the variation-aware asymmetric activity migration, for which the on-chip temperatures are observably reduced (compared to Figure 1).

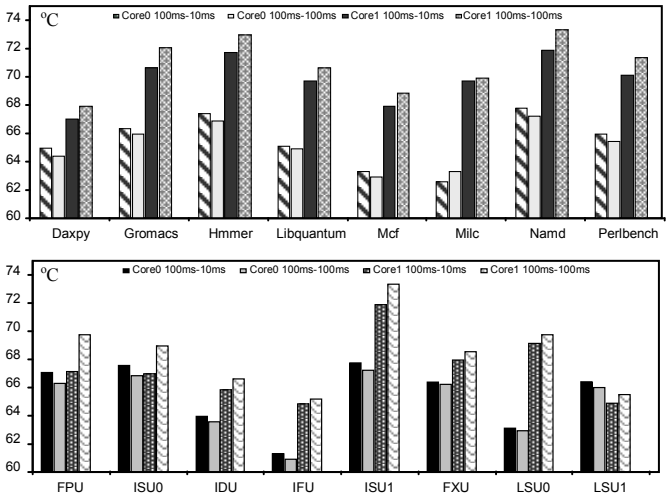


Figure 7. (a) Peak temperatures over SPEC2k6; (b) Peak block temperatures for core0 and core1 in SMT2 mode

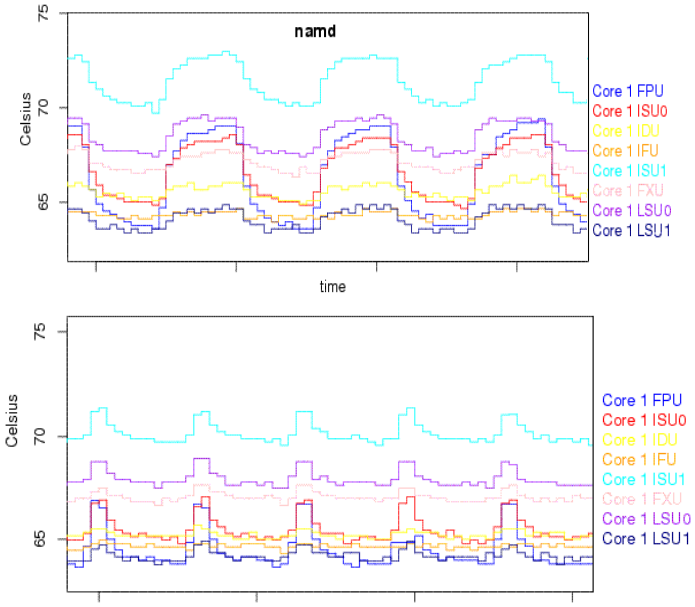


Figure 8. Effect of activity migration intervals on block temperature (Hotspot ISU - in light blue; block temperatures color coded for best viewing)

Figure 10 illustrates the heating behavior in time. In the top part of the figure '100ms-100ms' symmetrical core hopping interval causes core 1 to heat up, compared to the '100ms-10ms' case, where the cooler core 0 is utilized more intensely to improve the thermal profile.

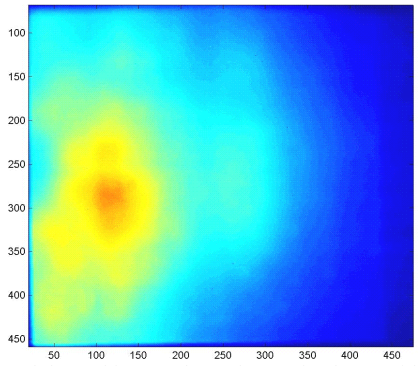


Figure 9. Infrared thermal image (thermal map) for daxpy with proposed variation-aware activity migration

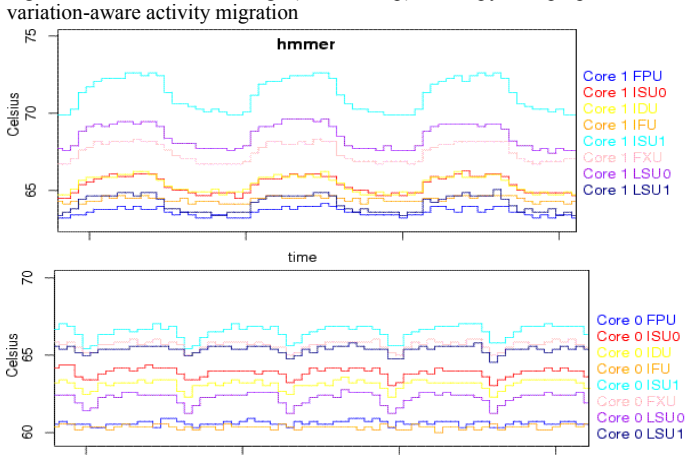


Figure 10. Temperature profile for hmmer 100-100ms intervals, 100-10ms migration intervals

D. Variation-aware Task Scheduling

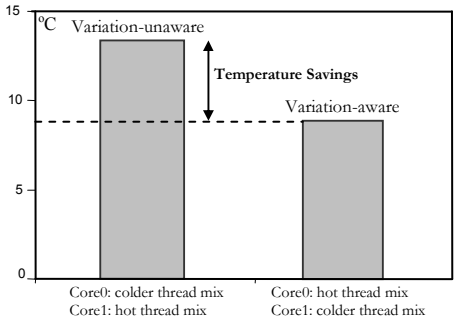


Figure 11. Peak temperatures (relative to linux idle loop) for alternative scheduling schemes: variation-unaware (left) and variation-aware (right)

To further demonstrate the advantages we implemented the proposed variation-awareness in thermal-aware task scheduling. In this experiment, we run workloads that utilize all four hardware threads on the chip. Figure 11 shows an example case, where the reduction in maximum chip temperature through variation-aware scheduling of hot threads from Spec2k6 (e.g. *namd)* and cold threads (e.g. *hmmer)* is illustrated. Without the assessment scheme, an oblivious task scheduler can assign the hot threads on the hotter core, resulting in higher peak temperature. The right bar shows that by using the proposed scheme the peak temperatures can be reduced by 4.5ºC. The two cores have the same IPC at the specified clock frequency. Therefore there is no performance cost for reducing the peak temperatures with variation-awareness. The experiment shows that by assessing a