RC24356 (W0709-107) September 26, 2007
Computer Science

IBM Research Report

Thermal Response to DVFS: Analysis with an Intel Pentium M

Heather Hanson, Steven W. Keckler
The University of Texas at Austin

Soraya Ghiasi, Karthick Rajamani, Freeman Rawson, Juan Rubio
IBM Research Division
Austin Research Laboratory
11501 Burnet Road
Austin, TX 78758

Thermal Response to DVFS:
Analysis with an Intel Pentium M

Heather Hanson
Dept. of Electrical & Computer Engineering
The University of Texas at Austin

Stephen W. Keckler
Dept. of Computer Sciences
The University of Texas at Austin

Soraya Ghiasi
IBM Austin Research Laboratory

Karthick Rajamani
IBM Austin Research Laboratory

Freeman Rawson
IBM Austin Research Laboratory

Juan Rubio
IBM Austin Research Laboratory

ABSTRACT

Increasing power density in computing systems from laptops to servers has spurred interest in dynamic thermal management. Based on the success of dynamic voltage and frequency scaling (DVFS) in managing power and energy, DVFS may be a viable option for thermal management, as well. However, publicly available data on the thermal effects of DVFS are very limited. In this work, we characterize the thermal response of Intel Pentium M system to DVFS, identifying the response timescale and influence of factors beyond voltage and frequency on processor temperature.

Categories and Subject Descriptors

B.0 [**Hardware**]: General

General Terms

Measurement, Experimentation

Keywords

temperature, DVFS, thermal measurement, thermal management, microprocessor

1. INTRODUCTION

Dynamic thermal management (DTM) is essential to computing systems, as the full spectrum from mobile devices to densely packed server racks faces serious temperature-related issues. Dynamic voltage and frequency scaling (DVFS) has been employed with great success for power and energy management [1, 4, 9] and shows promise for thermal management, as well.

Most modern microprocessors are equipped with DVFS to implement the Advanced Computer Power Interface (ACPI) performance states, or p-states, as well as thermal sensors for the purpose of hardware or software-controlled CPU temperature regulation [2, 8, 13]. In fact, the Pentium M was designed to use DVFS as a thermal throttling mechanism to avoid catastrophic temperature levels [5]. However, additional factors beyond DVFS influence CPU temperature: air flow, heat sink design, altitude, proximity to other heat sources, ambient temperature, workload activity, and others. These external factors vary by system installation, and vary through time for a given system.

In a survey of multi-core dynamic thermal management including DVFS, Donald *et al.* provide CPU temperature measurements for SPEC CPU2000 programs for one p-state (DVFS setting) on a Pentium M laptop computer [3]. They observed while some benchmarks converged to steady-state temperatures, others fluctuated. We performed similar experiments and observed that even benchmarks that reached steady-state temperatures did not converge to the *same* temperature each time. If DVFS does not directly control CPU temperature, how would the DVFS mechanism perform for DTM?

While empirical data for a detailed study of the thermal response to DVFS may be accessible within industrial settings, such data are not widely available to the research community. We offer the results of our study to share a new set of data points on one specific platform and to discuss common issues critical to DTM. In this paper, we address the following questions:

- What is the impact of DVFS p-states on temperature?
- What is the timescale of thermal response to p-state changes?
- What is the relationship between power and temperature across DVFS states?
- Would a simple thermal estimation model provide sufficient accuracy to use in DTM?

We track the transient and steady-state thermal responses to each DVFS state with custom microbenchmarks. We observe a two-stage response for CPU temperature after a DVFS p-state change: first, an initial exponential response to thermal plateau (most of the temperature change

Frequency (MHz)	Voltage (V)
2000	1.340
1800	1.292
1600	1.244
1400	1.196
1200	1.148
1000	1.100
800	1.052
600	0.988

Table 1: P-States

Figure 1: Pentium M (left) and data acquisition (right)

is within the first 100 ms, with another minute or so to fully reach the plateau), then a slower drift over a few minutes to a second plateau due to the effect of the local air temperature responding to the changing CPU temperature. In addition to characterizing the transient response to p-state changes with microbenchmarks, we use the SPEC CPU2000 benchmark suite to understand the joint impact of p-state and workload activity on processor temperature. We find that temperature varies by 17 °C throughout the suite at the maximum frequency and voltage p-state, a significant swing from a low of 38 °C to a high of 55 °C.

Section 2 discusses the Pentium M system and data acquisition methodology used in the experiments. Section 3 presents measured power and thermal response to p-state and workload changes, and the paper concludes with final observations in Section 4.

2. METHODOLOGY

2.1 Pentium M

The Pentium M 755 desktop processor system, shown on the left in Figure 1, consists of a single-core Dothan 90-nm processor supported by a Foxconn heat-sink and fan-assembly, an Intel 855GME chipset, 512 MB of DDR SDRAM memory and Radisys uniprocessor motherboard in a desktop form factor [11]. The operating system is Red Hat Linux Enterprise 4.

The processor supports 8 frequency-voltage pairs, listed in Table 1, from 600 MHz to 2.0 GHz. We use the most conservative voltage settings, VID#A in the processor datasheet, which range from 0.988V to 1.340V [7]. We refer to these 8 p-states by their frequency values, noting that each frequency is paired with a unique corresponding voltage. Changing the DVFS setting incurs a stall of up to 500μsec. Extensive clock gating produces a wide variation in processor power and temperature within a single p-state, according to workload activity.

2.2 Power Measurement

Two voltage-regulator modules supply power to the Pentium M processor. Voltage probes measure voltage drop across high-precision resistors inserted between each voltage-regulator module and the processor, sending values for the voltage drop and the processor core voltage to a National Instruments data acquisition system that records the voltage and calculates supply current, shown on the right in Figure 1. A custom virtual oscilloscope in LabView software displays sensor information and sends packets of measured current and voltage to the Pentium M over a network connection. We created customized drivers for the Pentium M to control DVFS settings and query power and temperature readings.

2.3 Temperature Measurement

We collect two temperature measurements: CPU temperature T_{CPU} and ambient temperature $T_{ambient}$. The CPU temperature sensor consists of an analog thermal diode located within the processor chip package and an A/D converter in the fan controller [13]. $T_{ambient}$ approximates the room ambient with an additional thermal diode and A/D converter on the motherboard that is exposed to ambient air. The recorded T_{CPU} values for this system are lower than in an enclosed environment such as a laptop, highlighting the importance in considering the cooling system in thermal analysis and simulations. Temperature values are recorded with a resolution of 1 °C and accuracy of +/- 3 °C [10].

2.4 Fan Control

The fan controller is typically configured to track CPU temperature, spinning faster at higher temperatures and more slowly when the chip is cooler. We customized the fan control for these experiments, turning the fan to a high rate of 4500 rpm for maximum cooling scenarios and zero rpm (off) to simulate harsh thermal conditions.

3. THERMAL CHARACTERIZATION

In this section, we present the highlights of our thermal analysis with microbenchmarks and the SPEC CPU2000 benchmark suite [6]. First, we recorded power and CPU temperatures for a series of 3 microbenchmarks. Each microbenchmark performs one task repeatedly, with a steady rate of workload activity. `daxpy` performs floating point adds and multiplies with very few level-1 cache misses, resulting in continuous high power consumption. `mcopy` copies data from one range of memory addresses to another, primarily within the level-2 cache, and exhibits a steady mid-range power consumption. `idle` is a low-power benchmark that is the unix `sleep` command applied for a fixed amount of time.

3.1 Transient Response

To capture transient thermal response, we recorded a continuous trace of the `daxpy` benchmark executing at each p-state for 200 seconds, from 2 GHz down to 600 MHz. The power measurement interval is 50ms per sample and temperature is queried every two power samples (due to a slower bus interface to query the fan controller), 100ms per unique temperature sample. In these experiments, the CPU fan is configured to spin continuously at the highest rate for maximum cooling.

Figure 2 plots the sharp drop in power with each step down in frequency. P-state changes are instantaneous at the measurement timescale. Power exhibits a clear relationship with DVFS setting. Figure 3 shows the corresponding trace of measured T_{CPU} and $T_{ambient}$ values. In most cases, T_{CPU} dropped by a total of 3 °C following a transition to a neighboring p-state (200 MHz, 50mV difference), regardless of frequency or power levels. The complementary case of ascending p-states exhibited similar behavior.

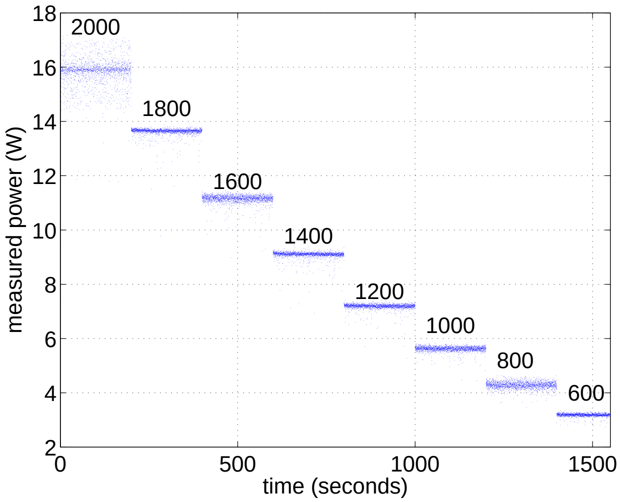

Figure 2: CPU power for `daxpy`, 200 seconds per p-state (denoted in MHz).

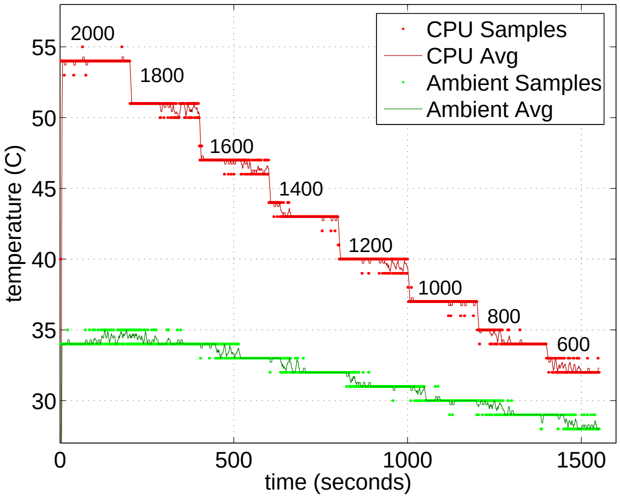

Figure 3: CPU and ambient temperatures for `daxpy`, 200 seconds per p-state (denoted in MHz).

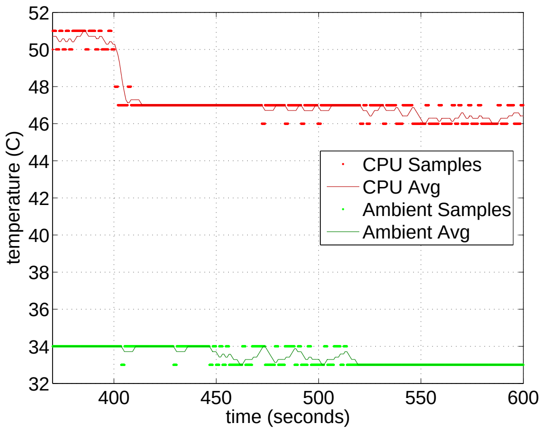

Figure 4: CPU temperature detail for `daxpy` p-state transition. Temperature adjusts in two stages: initial drop, then additional drift as ambient settles.

Figure 4 shows a closeup view of T_{CPU} during the transition from 1600 MHz to 1400 MHz p-states. Each p-state change causes an initial drop in T_{CPU} of 1-3 degrees. The initial drop is exponential in shape, with most of the temperature delta within the first sampling interval and a longer 'tail' of another degree within the first minute after a p-state change. Then, the influence of the gradually changing ambient temperature is noticeable as the CPU temperature continues to cool slightly, up to 1 more degree, when the local air temperature detected by the ambient sensor has reached a plateau for the new p-state, about 1-2 minutes after the p-state transition. The initial exponential curve due to the thermal time constant is well known; we did not expect the second plateau due to the local ambient temperature drift. The small temperature difference between plateaus will not likely affect the choice of p-state, although the longer time to reach the final plateau may affect the settling time for DTM with closed-loop feedback.

3.2 Steady-State Response

To gauge steady-state response, we executed microbenchmarks twice consecutively at each p-state. During the first run, temperatures transition from initial conditions to a steady temperature and in the second run, temperatures maintain their steady-state level. Each instance executed for at least 10 minutes, while the CPU fan spun continuously for maximum cooling.

Figure 5 plots the mean CPU power and temperature from the steady-state run for each microbenchmark, at each p-state. The data indicate that p-state alone does not dictate power or temperature: note the spread between `daxpy` and `idle` at the same p-state, due to clock gating and workload activity levels. For a given steady workload, however, both power and temperature scale with p-state under maximum-cooled conditions.

A linear relationship between power and temperature is evident in Figure 6. The slight variation in slope for each benchmark is most likely due to temperature sensor placement relative to workload-specific hotspots on the processor; the single sensor may be closer to `daxpy`'s hotspots than `mcopy`'s. Additional sensors on-die would give a more complete picture of hotspots and workload-dependent power-thermal relationships; a single measurement point provides insight to the overall thermal response to DVFS.

3.3 Environmental Influences

The transient response experiment demonstrated the interaction between CPU and ambient temperatures while the p-state stepped down in frequency. We investigated further to observe the effects of the processor's thermal environment on the T_{CPU}, analyzing the behavior of under-cooled systems and the effects of variable ambient temperature.

In an experiment to observe thermal response in an under-cooled system, we executed microbenchmarks while the fan was disabled. Figure 7 shows an experiment with stepped p-state levels for the `daxpy` benchmark. Leakage current is exponentially dependent on temperature; higher temperatures produce higher leakage current and greater power consumption. Power and temperature can exhibit a feedback effect

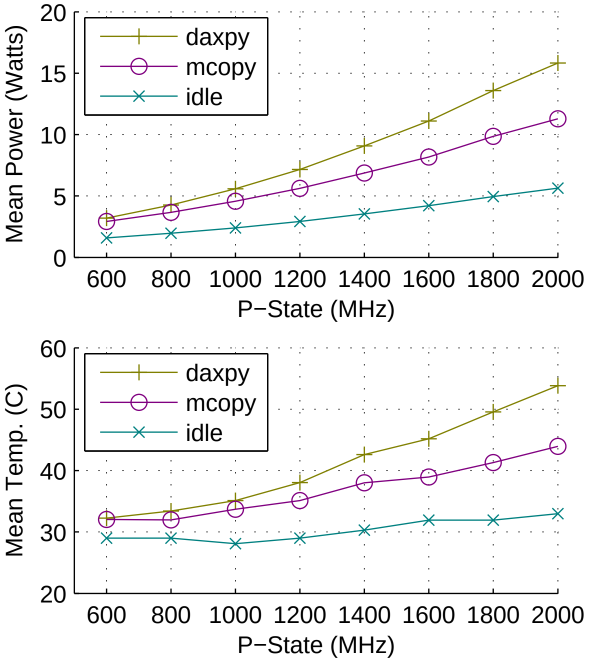

Figure 5: CPU power and temperature vary by benchmark. Power closely tracks p-state; CPU temperature loosely tracks p-state for given benchmark.

of increasing temperatures raising leakage current, in turn increasing power consumption, which generates more heat and further raises the temperature. The thermal runaway feedback effect is more pronounced at power levels above 10 Watts in this experiment. We expect that the system is better able to dissipate the extra heat generated from leakage current during lower power levels (lower total heat output), reducing the effect of leakage power on T_{CPU} and thus attenuating the feedback effect.

To study the effects of drifting ambient temperature over a long time period, we repeated the `daxpy` benchmark at each p-state from 2000 MHz down to 600 MHz, executing the set of 8 fixed p-states ten times, for a total of 80 invocations that ran continuously for approximately 23.5 hours. Figure 8 shows the minimum, mean, and maximum power and temperatures. Although power variation over repeated instances at the same p-state is negligible, measured temperatures vary by 5 °C. We investigated the cause of thermal variation for these steady-behavior microbenchmarks, and determined that during this test, a combination of external weather conditions and building heating/cooling settings caused the ambient temperature to drift by about 5 °C, causing a thermal offset for T_{CPU}.

3.4 Realistic Workloads

For a view of the thermal response with more typical workloads, we executed the full SPEC CPU2000 (floating-point and integer) suite with a fixed p-state for the duration of the run, for each of the 8 p-states, with maximum-cooling conditions.

Figure 9 illustrates the effect of p-states on power and temperature for one high-activity benchmark, `galgel`. The

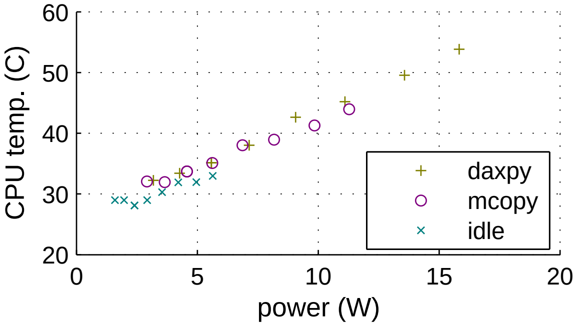

Figure 6: Linear power-thermal relationship under steady-state conditions for single instance of each benchmark and frequency.

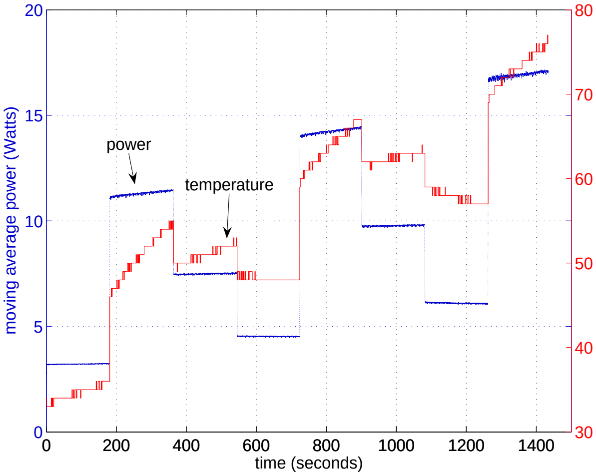

Figure 7: Power and temperature for `daxpy` benchmark with under-cooled conditions, with 200 seconds of each p-state in order: 600, 1600, 1200, 800, 1800, 1400, 1000, 2000 MHz.

benchmark executed in its entirety for each fixed p-state; all eight p-states are plotted in the figure from top (2000 MHz) to bottom (600 MHz). Workload characteristics can vary by p-state. `Galgel` exhibits periodic power swings with a distinctive zig-zag power pattern at higher frequencies during one phase of the benchmark. Since the memory speeds are unchanged with DVFS, the processor stalls for fewer cycles at lower frequencies, attenuating the bursty behavior observed at higher frequencies. Frequency-independent power, approximately 2-3 Watts, dominates the total power. As a result, the zig-zag power pattern is less noticeable at the low end of the frequency range, and nearly non-existent at the lowest p-state. T_{CPU} recorded for `galgel` reflects the power trends, fluctuating at high frequencies while maintaining a steady temperature (within the sensor resolution) at low frequencies. T_{CPU} values range from 32 °C to 56 °C, similar to the range recorded for microbenchmarks.

We executed the full SPEC CPU2000 suite for each of the eight p-states under maximum cooling conditions. Figure 10 plots mean power and CPU temperature for each benchmark

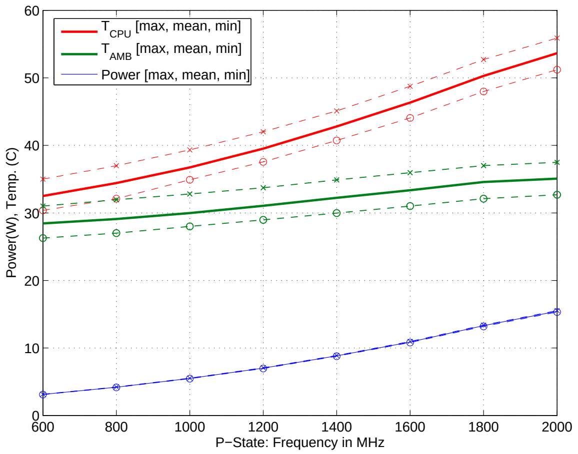

Figure 8: Steady-state power and temperature measurements for multiple invocations of `daxpy`.

from `gzip` through `apsi`, in SPEC execution order. The highest- and lowest-frequency p-states include vertical bars to indicate minimum and maximum recorded values within each benchmark. Temperature variation is larger for higher frequencies than lower frequencies, with greater minimum-maximum ranges and also larger differences between benchmarks' mean temperatures. Workload characteristics also influence T_{CPU}, as the benchmark `mcf` at the 2000 MHz p-state exhibits a mean temperature similar to `crafty` at 1600 MHz.

3.5 Thermal Model

We applied our observations of thermal response to DVFS to develop a thermal estimation model that predicts the CPU temperature response to changing p-states based on current conditions, for use in a power-temperature controller. We applied linear regression to the empirical steady-state ambient and CPU temperatures and power for microbenchmarks measured at each p-state to create a thermal model. The model captures the effects of both environmental conditions and power consumption on the CPU temperature:

$$T_{CPUest} = \tau P + T_{ambient} \qquad (1)$$

where T_{CPUest} is the estimated CPU temperature, τ is a scalar coefficient, P is the processor power at a given p-state, and $T_{ambient}$ is the current ambient temperature. Linear regressions indicate that the coefficient τ varies slightly by benchmark; we surmise that the difference is due to the single CPU thermal sensor that is spatially closer to the hotspots of some workloads than others. In our work, we simplify the equation to use a fixed constant of $\tau = 1.25$.

Other forms of a predictive thermal model would also be possible, such as directly predicting CPU temperature for other p-states given the current CPU temperature. The form of Equation 1 proved useful by leveraging our prior work that estimates power at all p-states based on measurements for the current p-state [12]. By using predicted power in Equation 1, we are able to quickly project CPU temperature for all p-states. The thermal model also exploits the slow rate of ambient temperature change. In systems with infrequent measurements or a long delay for temperature sensor readings, a slow-moving reference point in the estimation model such as the ambient temperature better tolerates sensor delay than a quickly changing measurement such as the CPU temperature.

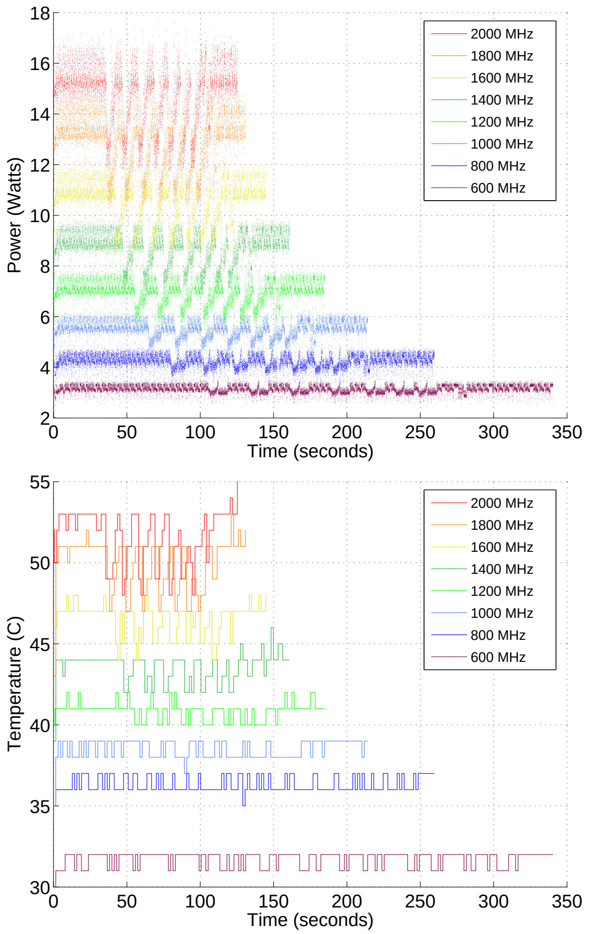

Figure 9: CPU power and temperature for `galgel` at each DVFS p-state.

Figure 11 charts each measured recorded data point of SPEC CPU2000 suite executing at 2 GHz with a corresponding temperature estimate based on power and ambient temperature (data points are vertically aligned due to integer measured values). The diagonal line represents a perfect prediction; above the line is an over-estimate and below the line is an under-estimate. The thermal model under-estimates in less than 5% of samples, with an average of 1.3 °C for underestimates. The model over-estimates T_{CPU} in 95% of all samples, with a mean of 3.4 °C for overestimates. The bias toward overestimates stems from the model training dataset of high-activity benchmarks that produce higher T_{CPU} values than the SPEC CPU2000 workloads, and is useful for situations that warrant a conservative estimate. More aggressive models could shift the error toward a more balanced over- and under-estimation and rely on the built-in thermal safety features in the event of a grave mis-prediction.

4. CONCLUSION

In this work, we characterize the thermal response of an Intel Pentium M system to DVFS.

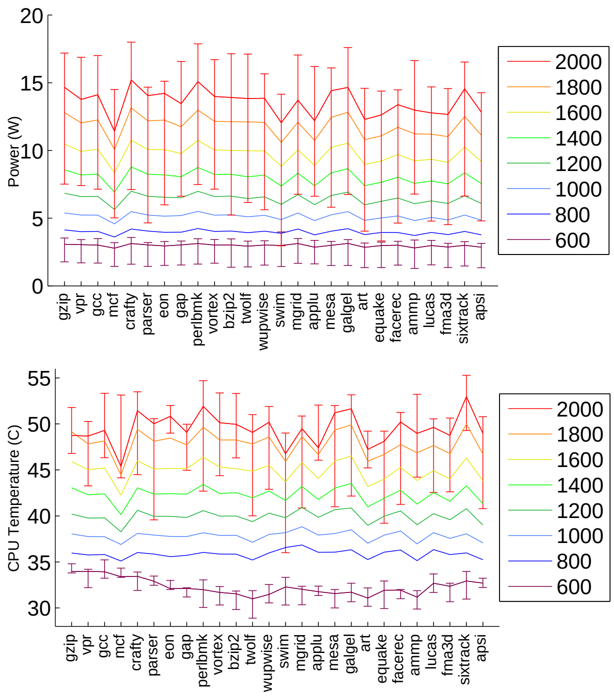

Figure 10: Mean and range of CPU power and temperature for each SPEC CPU2000 benchmark, at each DVFS p-state.

- We demonstrate that CPU temperatures scale with DVFS p-states under well-cooled conditions, for a given workload activity and ambient temperature.
- We identify the two-stage thermal response to p-state changes: a quick thermal change (milliseconds) followed by additional drift after the local air temperature adjusts to the new CPU temperature (minutes).
- We demonstrate a linear relationship between power and temperature, in a well-cooled environment.
- We develop a simple thermal estimation model based on current observed conditions to predict the effect of DVFS options on CPU temperature.

Our experiments show that DVFS has an immediate influence on processor temperature and confirm that DVFS could be a viable thermal control mechanism. However, CPU temperature is also affected by other factors, including workload activity and cooling capacity, thus highlighting the need for accurate and timely thermal sensor data to reflect current conditions for use in dynamic thermal management.

5. ACKNOWLEDGMENTS

This research was supported financially by an IBM Faculty Partnership award and by the Defense Advanced Research Projects Agency under contracts F33615-01-C-1892 and NBCH30390004.

6. REFERENCES

[1] Advanced Micro Devices. PowerNow with optimized power management, Jan. 2006.

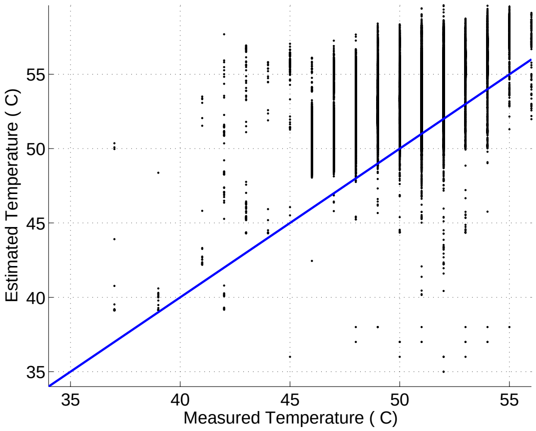

Figure 11: Comparison of estimated and measured CPU temperature.

[2] J. Clabes et al. Design and implementation of the POWER5 microprocessor. In *Design Automation Conference (DAC)*, pages 670–672, 2004.

[3] J. Donald and M. Martonosi. Techniques for multicore thermal management: Classification and new exploration. In *International Symposium on Computer Architecture (ISCA)*, pages 78–88, 2006.

[4] K. Flautner and T. Mudge. Vertigo: Automatic performance-setting for linux. In *Operating Systems Design and Implementation (OSDI)*, pages 105–116, 2002.

[5] D. Genossar and N. Shamir. Intel Pentium M power estimation, budgeting, optimization, and validation. *Intel Technology Journal*, 7(2):44–49, May 2003.

[6] H. Hanson. *Coordinated Power, Energy, and Temperature Management*. PhD thesis, The University of Texas at Austin, 2007.

[7] Intel. Pentium M processor on 90 nm process with 2-MB L2 cache datasheet, January 2005.

[8] Intel Pentium 4 processor with 512-kb L2 cache on 0.13 micron process and Intel Pentium 4 processor extreme edition supporting hyper-threading technology datasheet.

[9] A. Nanduri. Dynamic Power Coordination. `http://www.intel.com/products/processor/coreduo/dynamicpowercoordination.htm`, 2006.

[10] LM85 hardware monitor with integrated fan control. `http://www.national.com/ds.cgi/LM/LM85.pdf`.

[11] Radisys Corporation. Endura LS855 product data sheet.

[12] K. Rajamani, H. Hanson, J. Rubio, S. Ghiasi, and F. Rawson. Application-aware power management. In *International Symposium on Workload Characterization*, pages 39–48, October 2006.

[13] E. Rotem, A. Naveh, M. Moffie, and A. Mendelson. Analysis of thermal monitor features of the Intel Pentium M processor. In *Workshop on Temperature Aware Computer Systems*, June 2004.