Power and Thermal Characteristics of a Pentium M System

Heather Hanson[*] and Stephen W. Keckler[+]
[*] Department of Electrical & Computer Engineering
[+] Department of Computer Sciences
The University of Texas at Austin

IBM Technical Contacts: Karthick Rajamani
Austin Research Laboratory Power-Aware Systems Group

February 22, 2007

In this work, we characterize the power and thermal characteristics of a Pentium M system in response to a widely available power-management mechanism, DVFS (dynamic voltage and frequency scaling) and cooling capacity. We observe transient and steady-state responses to changing the DVFS state with custom microbenchmarks and develop a model that captures the power-thermal relationship for this system. We also record the power and thermal response to the SPEC CPU2000 benchmark suite executing at 2 GHz to determine the effects of a realistic workload with fluctuating power consumption. The measured data from live hardware illustrate the complex interaction between CPU and ambient temperatures and provides insight for power and temperature management by identifying the magnitude and timescale of power and thermal responses to control settings.

1 Introduction

Dynamic power and thermal management are essential to computing systems, as the full spectrum from mobile devices to densely packed server racks face serious power-related issues. System solutions to power and thermal management vary to some degree: server rooms require industrial-strength air conditioning, while notebook computers must operate with a small and reasonably quiet fan, yet microprocessors in diverse settings face common problems of current supply, power density, and heat dissipation. Controlling CPU temperature is a critical task, yet thermal data are not widely available to the research community.

In this paper, we characterize the power and thermal characteristics of a popular processor that is used in system from laptops to servers. We measure the system response to DVFS, cooling capacity, and workload changes to answer the following questions:

- What are the timescale and magnitude of thermal response to DVFS settings?
- What is the relationship between power and CPU temperature?

Frequency	Voltage
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: Pentium M DVFS p-states: Frequency and Voltage Pairs

Figure 1: Equipment: Pentium M on left, data acquisition module and measurement PC with virtual oscilloscope on right

- Can we use DVFS to control temperature?

We observe the transient and steady-state responses to each DVFS state with custom microbenchmarks. We also record the full SPEC CPU2000 benchmark suite executing at maximum speed to determine the effects of a realistic workload with fluctuating power consumption. Measured data from live hardware illustrate the complex interaction between power and temperature and provide insight for future work in power and thermal management by identifying the magnitude and timescale of responses to power-management settings and cooling capacity.

Section 2 discusses the Pentium M and data acquisition systems and the workloads used in the experiments. Section 3 presents measured power and thermal response to p-states for a series of microbenchmarks and Section 4 shows the power and thermal response of a realistic workload, the SPEC CPU2000 suite, to each p-state. The paper concludes with final observations in Section 5.

2 Methodology

2.1 Pentium M

The Pentium M system consists of a Pentium M 755 desktop processor system: a single-core “Dothan” series 90-nm processor supported by a Foxcon heat-sink and fan-assembly, an Intel 855GME chipset, 512 MB of DDR SDRAM memory and Radisys uni-processor motherboard [6]. The motherboard resides in a conventional PC enclosure, with top panel removed to allow access for probe cables.

The Pentium M follows the Advanced Configuration and Power Interface (ACPI) specification with multiple performance states, p-states, from P0 through Pn. **P0** is defined as the state that "uses its maximum performance capability and may consume maximum power", and **Pn** as the state with performance "at its minimum level and consumes minimal power while remaining in an active state" [1]. The Pentium M employs two p-state mechanisms: clock throttling and dynamic voltage and frequency scaling (DVFS) [4]. Clock throttling intermittently stops the clock for a specified percentage of the time. We previously studied clock throttling alone and in combination with DVFS and found that for this system, DVFS provided superior power-performance behavior. The rare situation that would warrant clock throttling would be to reduce power below the 2 watts that the 600 MHz p-state could provide. For a comparison of clock throttling and DVFS on this Pentium M system, refer to [2]. In this study, we use only DVFS p-states. DVFS supports 8 frequency-voltage pairs listed in Table 1, with 200 MHz steps from 600 MHz to 2.0 GHz, and voltages corresponding to the most conservative settings, VID#A in the processor datasheet [3]. We label p-states by their frequency throughout this document, noting that the p-state specifies both frequency and voltage (a frequency change is always accompanied by a voltage change). Changing the DVFS setting incurs a stall on the order of 1 mV/μsec [7], effectively instantaneous at millisecond sampling time scales.

2.2 Sensors

We tapped the high-precision resistors between each of voltage regulator modules and the processor with a voltage probe, providing voltages to a National Instruments (NI) data acquisition system that monitors processor supply voltage and also calculates supply current (via voltage drop across the sense resistors). A custom virtual oscilloscope in NI LabView software displays voltage and current information and sends UDP packets of measured data to the Pentium M.

We created customized drivers for the Pentium M to monitor performance, power, and temperature and to control DVFS settings. The power and performance sampling interval is nominally 100 samples per second, although the actual sampling interval length varies slightly, with most samples within 10-15 ms and a mean sample length of 13 ms, with an effective sampling rate of 80 samples per second on average. Power samples arrive in UDP packets at a rate of 100 samples per second. Due to the difference in sampling rates, the data acquisition system delivers more packets than the custom driver will consume; excess packets are discarded when the packet buffer becomes full. For steady-state conditions such as those in this study, the effect is negligible. When the workload behavior or p-state changes rapidly, it is possible for the power and performance samples to be misaligned. In those situations, the sampling rates can be tuned such that the UDP injection rate more closely matches the effective rate of the custom driver software.

A thermal diode in the processor package is connected to an external A/D (analog-to-digital) converter that translates the measurements into junction temperature. The Pentium M uses the temperature readings in two ways. An automatic safety feature reacts quickly if the junction temperature exceeds a preset threshold, enabling either DVFS or clock-throttling management options. A separate option is available through software control to read the junction temperature and respond with p-state changes according to user-defined policies. The automatic self-throttling feature will engage as needed even during software-controlled thermal management [7]. In our study, we use the software-controlled option, maintaining temperatures below the point at which the automatic safety feature would override our p-state choices.

In our system, reading temperature via a custom driver to the LM85 fan controller chip (that contains the A/D) is slow, and we found that waiting for temperature readings delayed the monitoring software. We decoupled the sampling rates for temperature, and allow one temperature sample per N power and performance samples. Typically, we use a rate of one temperature query per ten performance/power samples. For detailed temperature analysis we use one temperature sample per 2 performance/power samples and reduce the power/performance sampling rate. We collect two temperature measurements: the CPU temperature, for which a sensor is located within the processor chip package, and the ambient temperature, which is the motherboard temperature near the fresh-air intake vent. Temperature values are measured at 1-C resolution.

The manager spawns a benchmark as a child process, with the highest user-level priority. During benchmark execution, the software reads Pentium M performance counters and generates an output file at the conclusion of benchmark execution that includes timestamps, counter values, power measurements, temperature, and fan speed measurements.

2.3 Benchmarks

We used two types of benchmarks in this study: three microbenchmarks with steady, well-defined behavior and a suite of typical programs. `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 memory address 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 Linux `sleep` command applied for a fixed amount of time; the processor is minimally occupied with background tasks, similar to realistic behavior between workloads or while waiting on user response in interactive applications. For typical workload behavior, we used the SPEC CPU2000 suite of integer and floating point benchmarks.

3 Observations

The system's thermal response is determined by two factors in opposition: heat generated by power dissipation, and heat conduction by the cooling system. In this section, we observe the effects of power on temperature under maximum cooling conditions and the effects of the cooling environment on CPU temperature under steady-power conditions, with steady microbenchmarks.

3.1 Power Influence on CPU Temperature

3.1.1 Transient Response

Figure 2 shows a continuous trace of the `daxpy` benchmark executing as the p-state changes every 200 seconds, starting at the maximum frequency-voltage pair at 2 GHz, descending in 200 MHz steps to 600 MHz. In this experiment, the monitoring software recorded power samples at a rate of 20 samples per second (50ms each), with one temperature sample for every 2 power samples (100ms between unique samples). Note the sharp drop in power with each step in frequency. Stall times on the order

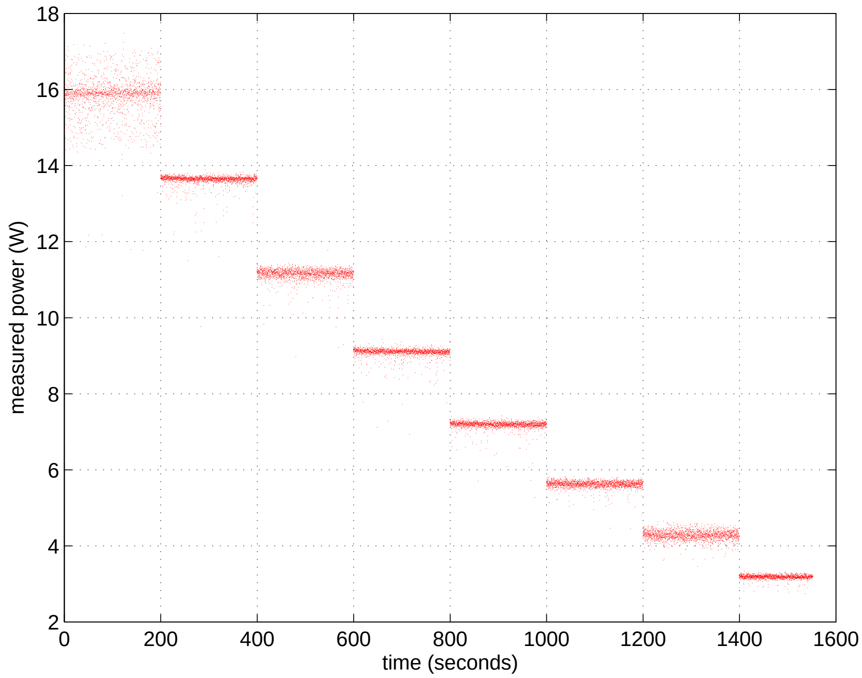


Figure 2: DAXPY power, 8 p-states.

of μs to change p-states are negligible compared to the sample length of 50 ms. Power exhibits a clear relationship with DVFS setting.

Figure 3 shows the corresponding trace of measured temperatures for the `daxpy` microbenchmark executing during descending frequency steps. Measured temperature bounces between integer values on the graph due to the coarse 1-degree resolution of the measurement. We observe that each p-state change to a lower frequency causes an initial sharp drop in CPU temperature within the first 50 ms sample, then additional descent of 1-3 degrees over several samples in most cases, followed by a longer period of slight decrease in temperature over the span of several seconds. Figure 4 shows two frequency changes, from 1800 MHz to 1600 MHz to 1400 MHz. After the change to 1600 MHz at the 400-second mark, the CPU temperature settles to 47 degrees Celsius within approximately 20 seconds and remains stable for about one minute, then begins a gradual, non-monotonic decrease of about 1 degree. At the 600-second mark, the p-state changes to 1400 MHz, at which point the temperature drops to 44 degrees, then settles to 43 degrees after approximately one minute.

In most cases, the CPU temperature dropped by 3 degrees following the p-state transition, typically with an immediate change of 2-3 degrees, with a longer 'tail' of 0-1 degrees throughout one minute after a p-state change. Then, the influence of the gradually changing ambient temperature is noticeable as the the CPU temperature continues to cool for another 1 degree over the next few minutes. Figure 3 shows both CPU and ambient temperatures decreasing over time. As the CPU temperature decreases, the ambient also decreases, which reduces the thermal load on the heatsink, which allows the CPU temperature to further decrease. At the low end of the frequency spectrum, when power is lowest and the cooling system is most capable of removing the CPU-generated heat from the system, the temperature decreased by total of 4 degrees after the transition from 800 MHz to 600 MHz.

Figures 5 and 6 show the complementary case of the `daxpy` benchmark and p-state steps with

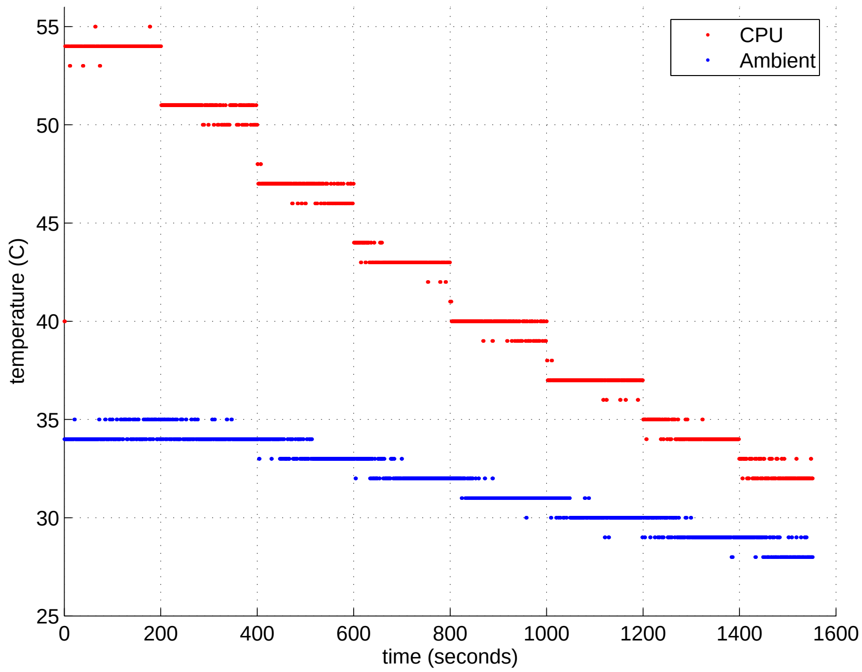


Figure 3: DAXPY CPU temperature, 8 p-states.

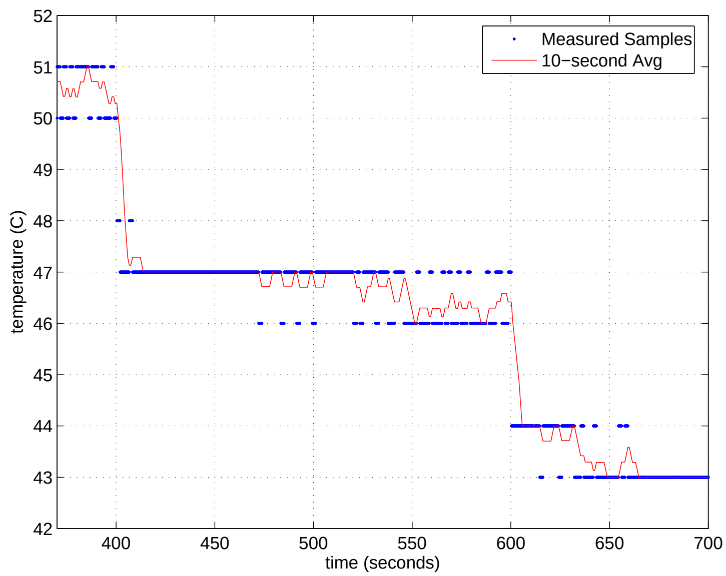


Figure 4: Frequency transitions 1800 MHz to 1600 MHz to 1400 MHz.

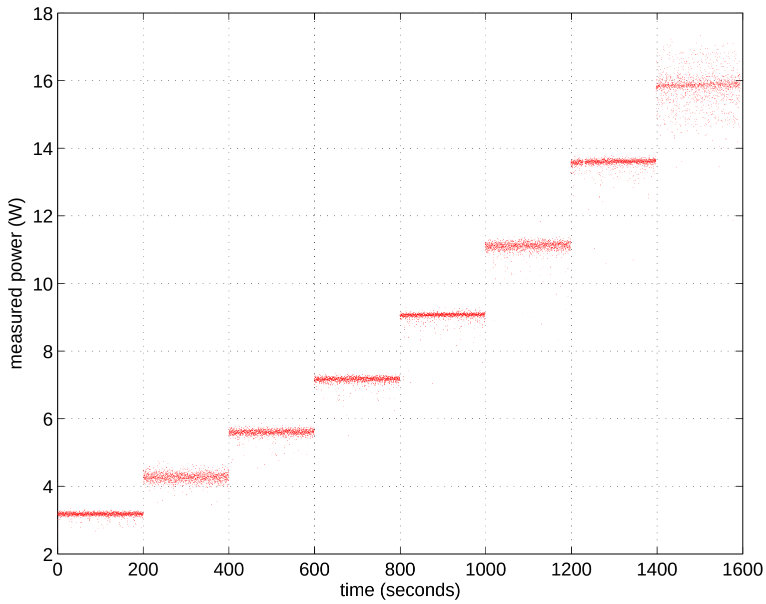


Figure 5: DAXPY Power with Ascending Frequencies, 2 GHz to 600 MHz in 200 MHz Steps.

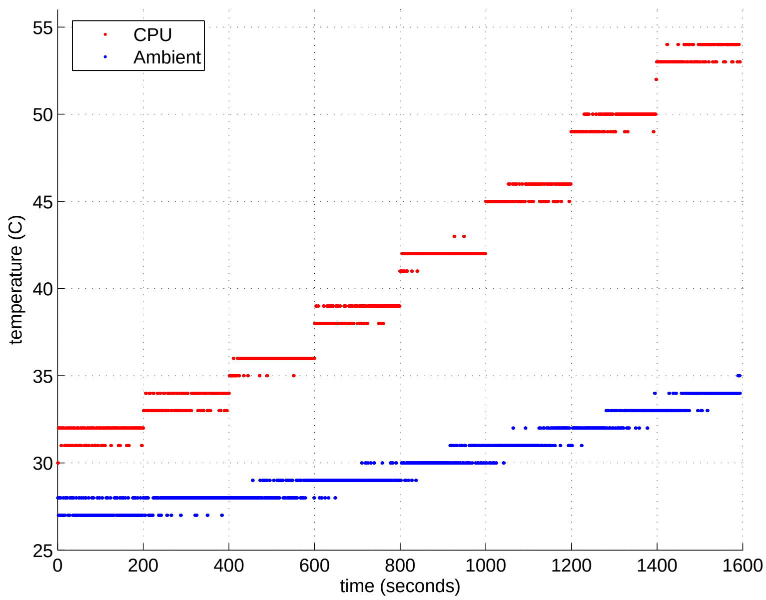


Figure 6: DAXPY CPU Temperature with Ascending Frequencies, 2 GHz to 600 MHz in 200 MHz Steps.

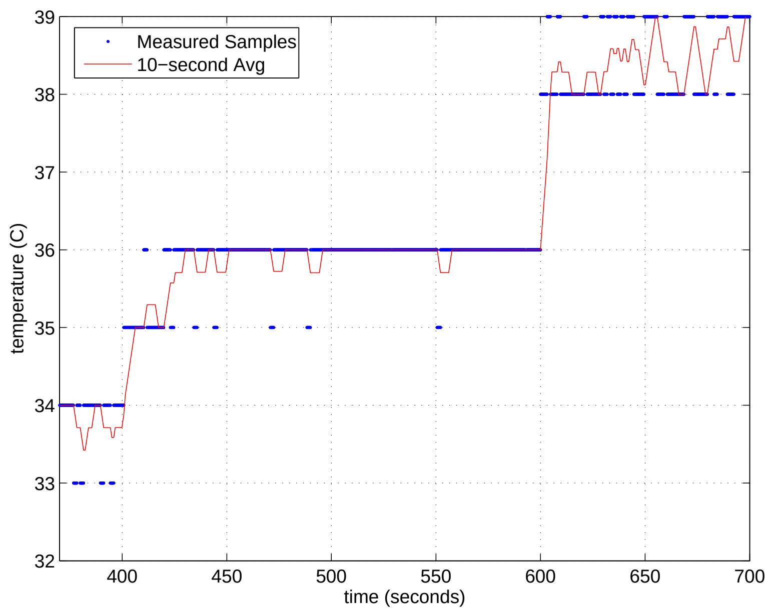


Figure 7: Frequency transitions 1800 MHz to 1600 MHz to 1400 MHz.

ascending frequencies. The mean power for each p-state is the same for ascending and descending frequencies, within 70 mW. Temperature is more variable, as it depends on the room conditions, including ambient temperature and airflow. The mean temperatures varied between the ascending and descending cases by up to 1.5 degrees. Like the descending case, measured power values form bands of steady behavior for the `daxpy` benchmark, with wider dense bands for 800 MHZ and 1600 MHz, and a wider, more sparse band for 2 GHz in Figure 5. Figure 6 shows how ambient and CPU temperature increase as the p-state ascends through all frequencies.

Figure 7 shows the timescale for ascending p-state transitions from 800 MHz to 1 GHz to 1.2 GHz, displaying individual measured points and a continuous line for the 10-second moving average. Like the descending case, the temperature initially changes a small amount quickly (1-3 degrees), then continues for another degree over about one minute. Figure 7 does not show a second-stage heating effect from the ambient in this portion of the trace; it is likely that the ambient influence created a temperature rise less than 1 degree Celsius, and thus is not measured with the thermal sensor's 1-degree resolution. As Figure 6 shows, the ambient does rise through time as the CPU temperature rises.

As an aside, note that the steady-state power levels form bands of varying thickness. The band for 2 GHz is the widest, at almost 3 watts, with individual points above and below the dense center near the mean. The other p-states have more outliers below the mean power and few outliers above the core power band. Both 1600 MHz and 800 MHz have thicker bands than other p-states; we observed this wider power variation at these frequencies in multiple experiments, and hypothesized that the wider power bands are related to a common clock distribution component for these related frequencies.

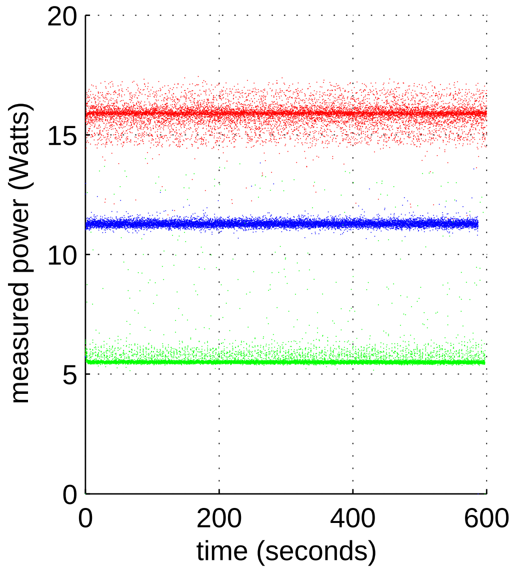


Figure 8: Microbenchmarks' steady-state power at 2 GHz.

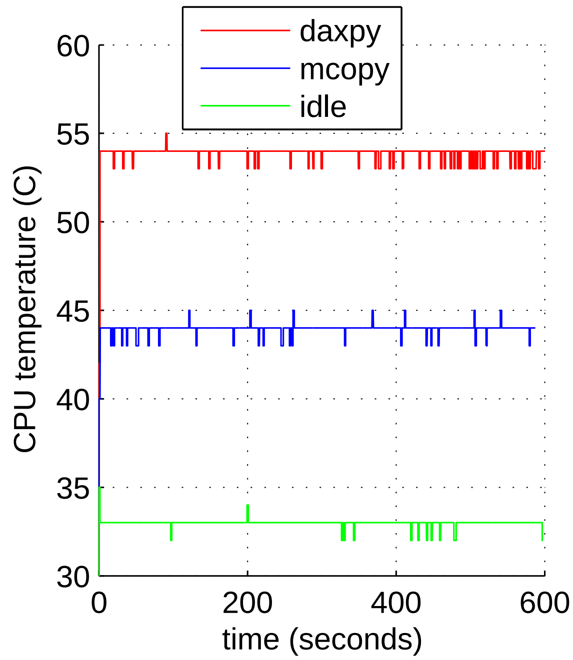


Figure 9: Microbenchmarks' steady-state CPU temperatures at 2 GHz.

3.1.2 Steady-State Response

Figures 8 and 9 show measured power and CPU temperatures, respectively, for a series of 3 microbenchmarks. The benchmarks each executed twice consecutively in the order `mcopy`, `daxpy`, and `idle`. The figure shows only the second execution of each benchmark, which captures the steady-state behavior. Each benchmark executed for approximately 10 minutes at a frequency of 2 GHz. In Figure 8, note that each benchmark maintains a steady band of power dissipation throughout its execution, and the power consumption varies by benchmark, even at the same p-state. In Figure 9, slight differences in actual temperature are amplified as values are rounded to the nearest full degree, resulting in the measured CPU temperatures varying by +/- 1 degree Celsius for steady-state behavior.

Figure 10 shows the standard deviation for power and CPU temperatures for each benchmark throughout the full range of p-states. Overall, the standard deviations are small, less than 1 Watt and 1 degree C, indicating that the steady-state measurements are indeed steady. `mcopy` and `daxpy` have well-defined continuous behavior running at a high priority, and for these two microbenchmarks, power standard deviation is generally low. The spread in measured temperatures is slightly higher. The coarse resolution of the temperature measurement magnifies small differences between integer values with rounding error, but also the temperature may actually have larger fluctuations due to high heat output relative to the cooling system capability.

The `idle` microbenchmark, on the other hand, is subject to the whims of the operating system and background processes, and is less steady in workload behavior and thus power throughout the range of p-states. However, the temperature spread is lower than the other benchmarks, possibly due in part to a steady-state temperature closer to an integer value, but it is also likely that the temperature is actually more steady due to the lower heat output from the CPU.

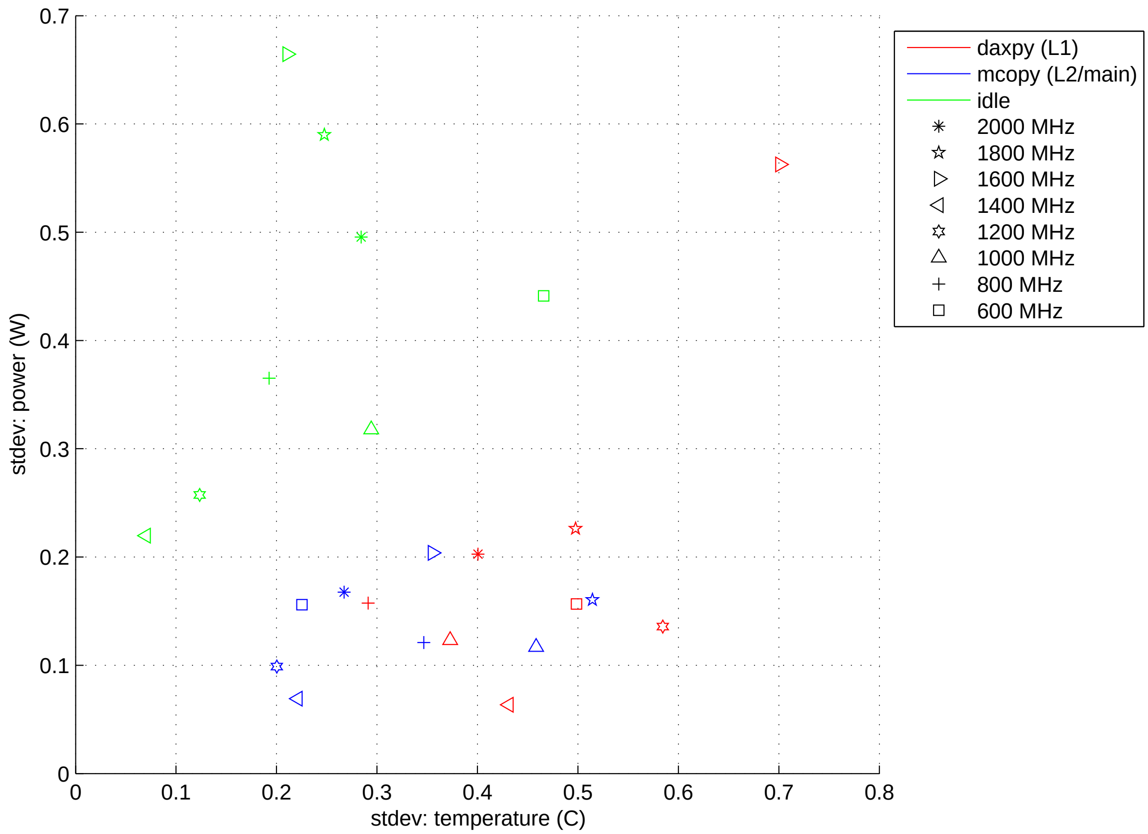


Figure 10: Standard Deviation for Measured Power and CPU Temperature

With the maximum fan speed, the lower power of `idle` is most easily managed by the cooling system; higher-power cases such as `daxpy` at high frequencies shows more thermal variation.

A fairly linear relationship between power and temperature is evident in Figure 11. Temperature shows a stronger relationship with power in the high-power regime, and a weaker coupling between power and temperature for lower-power points. We interpret these results to indicate that at high-power conditions, heat generation from the power dissipation dominates CPU temperature, while during low-power conditions, the effects of small differences in air flow, ambient temperatures, etc. are more evident in measured CPU temperature.

The slightly different slopes for `daxpy` and `mcopy` points in Figure 11 are 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, for example. Additional sensors on-die would give a more complete picture of the power-thermal relationship, however, even a single measurement point provides an indication of expected behavior, within a few degrees C.

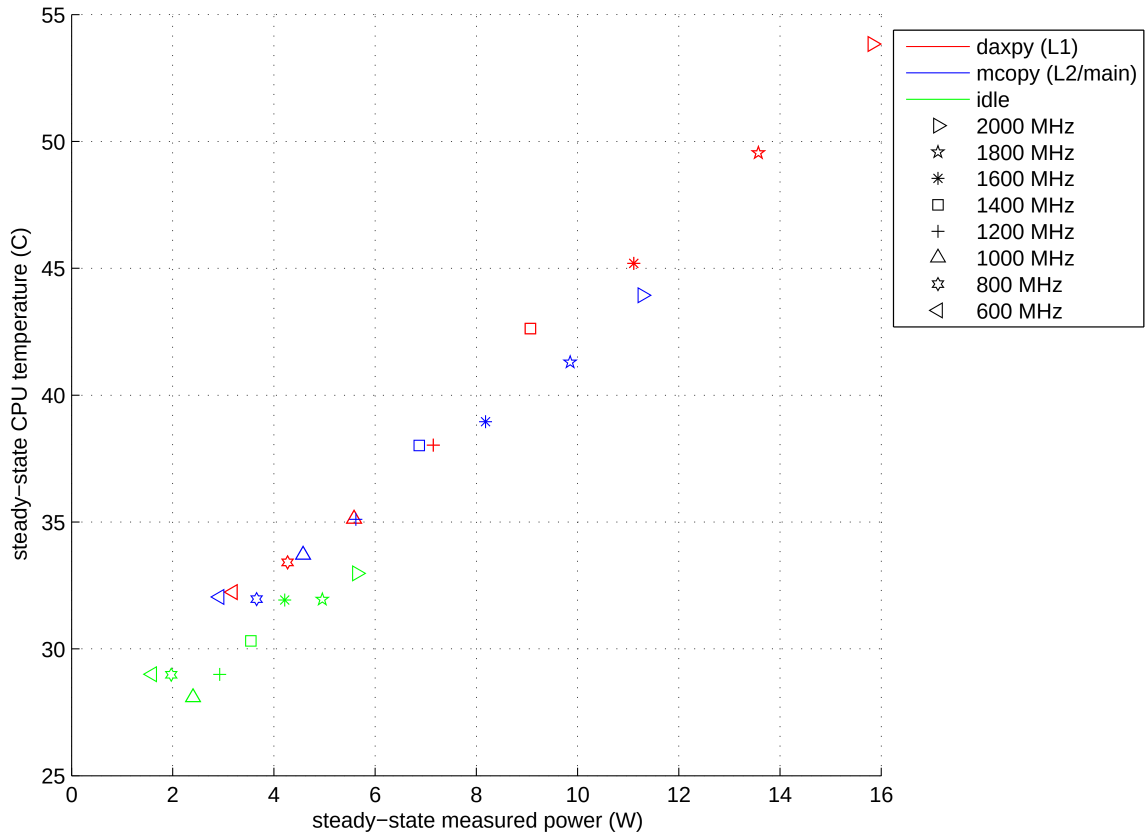


Figure 11: Mean power vs mean temperature for 3 microbenchmarks at 8 DVFS p-states.

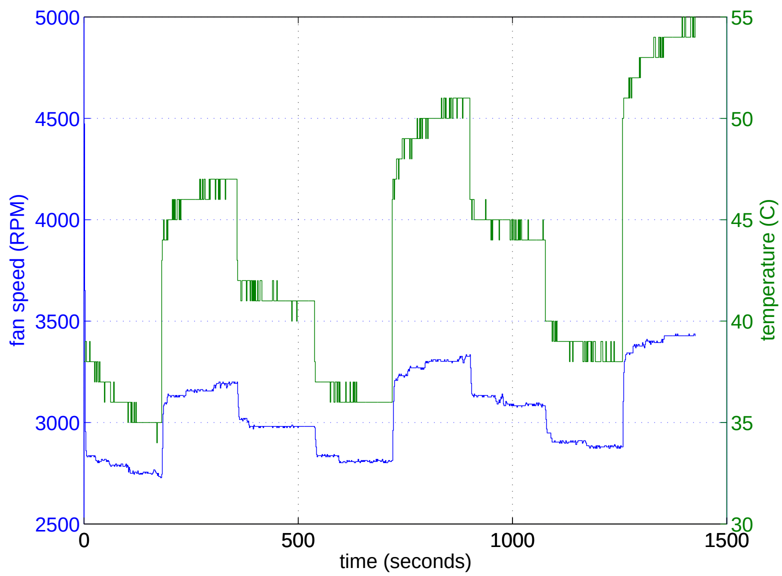


Figure 12: P-state changes in 400 MHz steps, with temperature-tracking fan

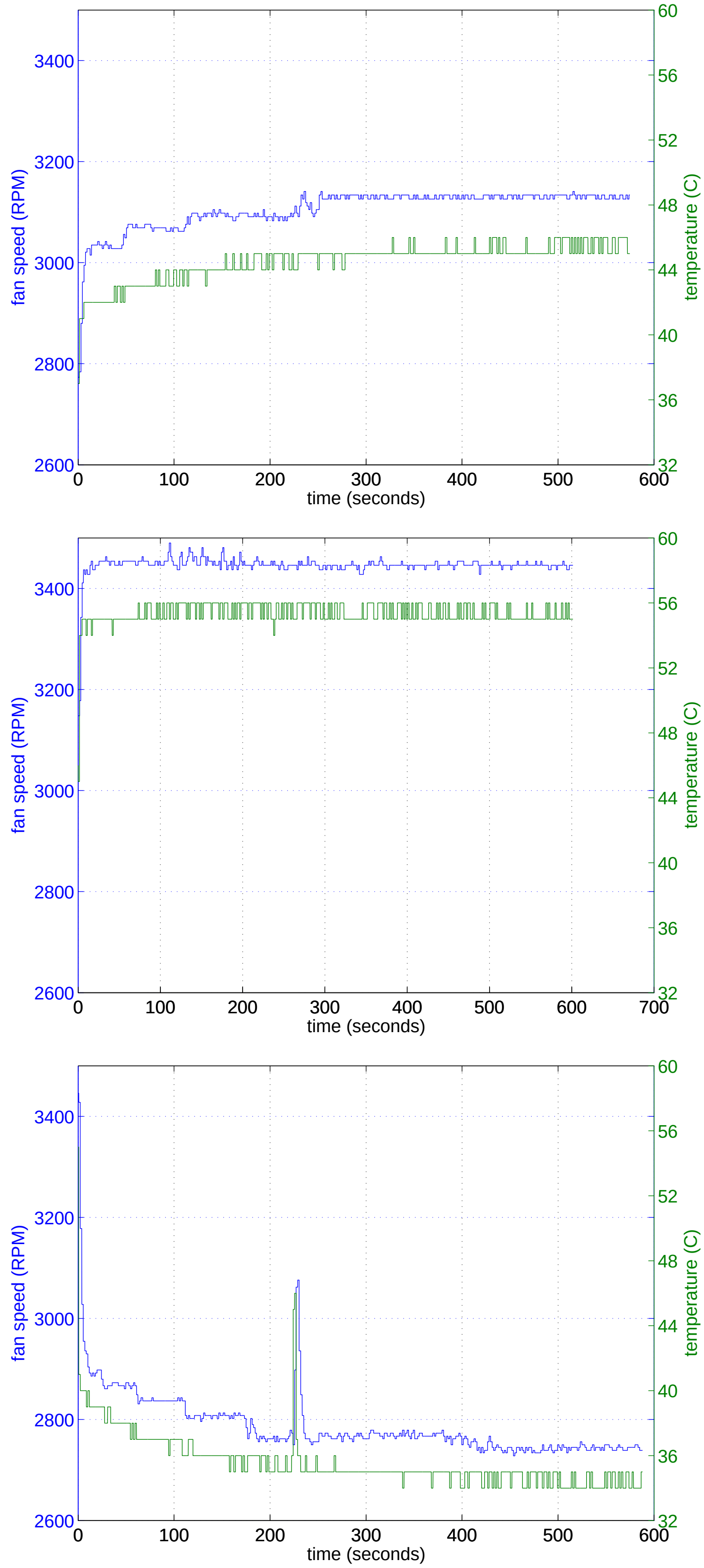
3.2 Cooling System Influence on Temperature

To analyze the effect of the cooling system on CPU temperature, we studied the system with three fan configurations: continuous maximum fan speed, a typical case of temperature-tracking fan, and with the fan turned off. The previous section explained the effect of power on temperature under maximum-speed fan conditions. This section presents experiments with typical and reduced cooling capacity.

3.2.1 Temperature-tracking Fan

Figure 12 shows the CPU temperature and fan speed as the p-state changes with large frequency differentials, increasing in 1 GHz steps and decreasing in 400 MHz steps. We observe similar thermal behavior as the previous transient response under maximum-fan conditions, with an initial sharp drop or rise, followed by a longer tail of temperature change.

Figure 13 shows the CPU temperature and fan speed for three microbenchmarks, executing at 2 GHz, while the fan operates under typical temperature-tracking conditions in which the fan speed is proportional to the CPU temperature. In this experiment, the benchmarks executed in the order of `mcopy`, `daxpy`, `idle`, at 2 GHz. `Mcopy` warms gradually to 45-46 degrees C, `daxpy` quickly reaches a steady level of 55-56 C, and `idle` cools to 36 C due to its lower power consumption. The time scale to reach steady-state temperatures varies from about 2 minutes for high-powered `daxpy`, about 5 minutes for `mcopy` and `idle`. The fan speeds reflect the temperature trends, with higher fan speeds for the hotter benchmarks. A short spike of activity during the `idle` test caused a temperature jump, followed by an increase in fan speed.



MCOPY

DAXPY

IDLE

Figure 13: Temperature-tracking fan: CPU temperature (green) and fan speed (blue). Due to different power levels, `mcopy` warms relatively slowly, `daxpy` warms quickly, and the system cools during the `idle`. The thermal spike in the `idle` test is the result of a short spurt of processor activity.

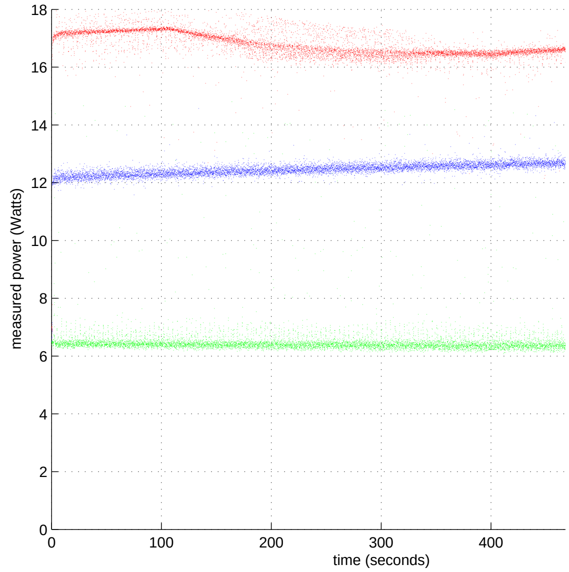


Figure 14: Disabled fan: CPU power for 3 microbenchmarks at 2 GHz

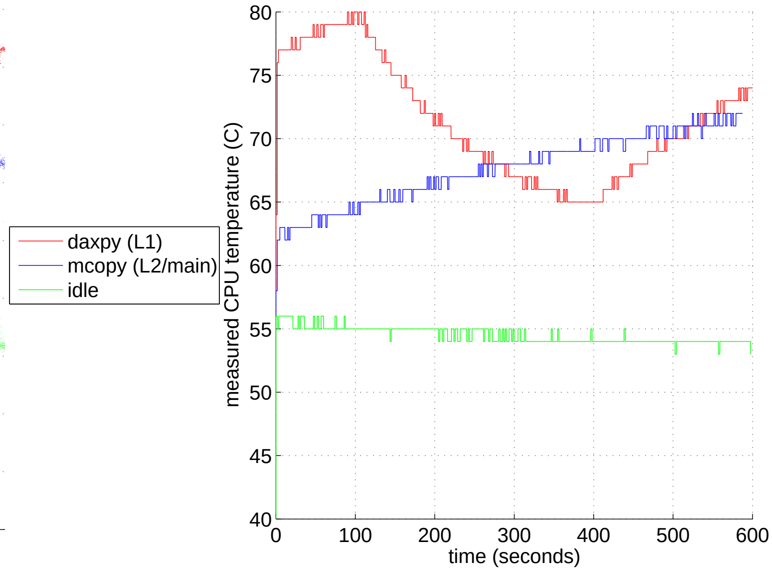


Figure 15: Disabled fan: CPU temperatures for 3 microbenchmarks at 2 GHz.

3.2.2 Disabled Fan

Figures 14 and 15 show the power and thermal measurements for each of the 3 benchmarks while the fan is disabled. These experiments are designed to show the Pentium M's response to the CPU fan turned off, both steady-state and transient power and thermal response, similar to the maximum and default fan cases. However, in this case, the 'disabled' fan did not remain disabled. Rather than directly disabling the fan, we set T_{min}, the temperature at which the fan turns on, to 80 C. In most cases, the Pentium M CPU operates will below 80 C and the fan remains off. However, if the CPU temperature does reach T_{min}, the fan turns on, as evident in the `daxpy` case. Note that the power and temperature for `daxpy` rise, dip, and rise again due to the fan operation. `Daxpy`'s high power consumption leads to high temperatures while the fan is disabled, quickly reaching the 80 C threshold for the fan to engage. The fan continues to spin at a low rate until the temperature decreases below a lower threshold, empirically observed to be 65 C in these experiments, which is 15 C below the upper threshold. The lower-powered benchmarks `mcopy` and `idle` do not exceed the 80-C limit and thus the fan remains disabled throughout their execution.

One interesting effect of disabling the fan is the difference between the `mcopy` and `idle` benchmarks. The CPU temperature rises for `mcopy` yet decreases slightly for the `idle`; power rises for `mcopy` in concert with the rising temperature, and remains steady for `idle`. The feedback effect of increasing power raising the temperature, in turn raising the power is evident only at higher power and temperatures. At lower power levels, the system remains cool enough to prevent the thermal feedback, even without the CPU fan's contribution to air flow. Figure 16 shows the effect in detail for the `daxpy` benchmark and a p-state change from 800 MHz to 1800 MHz while the CPU fan was disabled. Both temperature and power are steady at 800 MHz, and after the p-state transition, both temperature and power continue increase. We found that with the fan disabled, power consumption below 10 Watts was steady and power above the 10-W threshold exhibited symptoms of thermal runaway.

While turning the fan off is an extreme case, it does demonstrate the system properties under high-

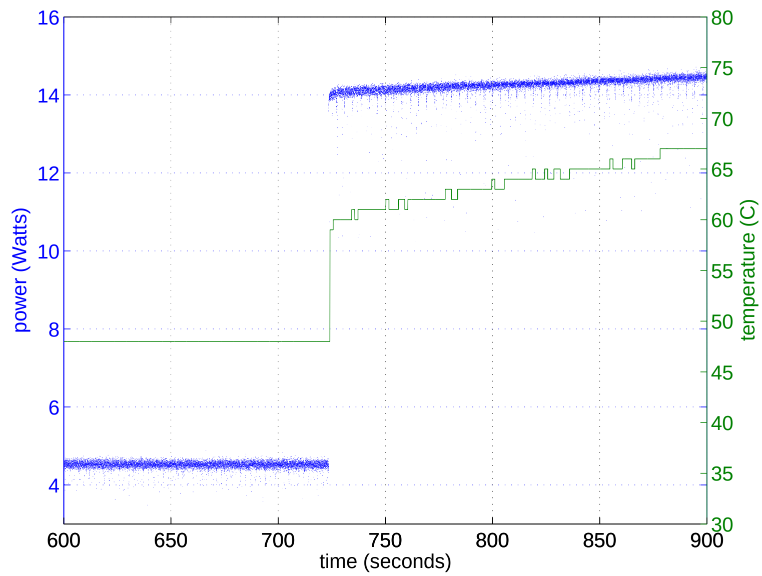


Figure 16: Disabled fan: p-state change from 800 MHz (steady) to 1800 MHz (thermal-power interaction) .

temperature conditions, such as those in the hot zones within server rooms or other under-cooled situations. Fans and blowers consume a large portion of server power budgets [5]. Understanding the effects of reducing the fan speed to zero is a first step toward managing overall system power by trading cooling capacity for power consumption.

3.3 Effect of Ambient Temperature

Figures 17 and 18 show the effect of the `daxpy` benchmark repeated 10 times for each p-state. The execution order was an inner loop of a single invocation of the benchmark for each p-state from the maximum frequency p-state down to the lowest-frequency p-state, and an outer loop of those 8 instances repeated 10 times. The benchmark repeated 80 times total from 5:40pm through 3pm the following day. External conditions caused the ambient temperature to vary by about 5 degrees, which has an effect on CPU temperature up to 5 degrees for the same steady workload and fixed p-state.

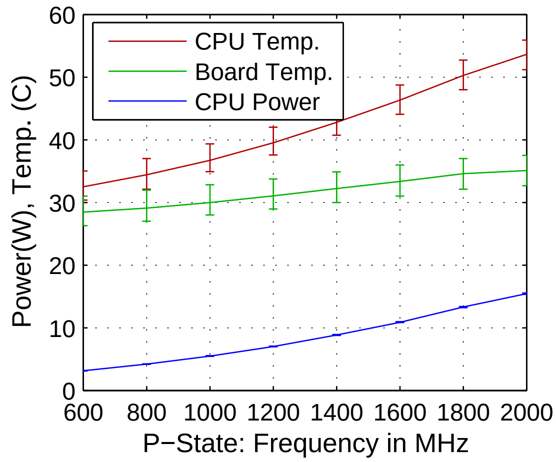


Figure 17: Power and temperature recorded for `daxpy` repeated 10x at each p-state. The mean values of power, CPU temperature, and ambient temperature are plotted for each p-state, along with minimum and maximum values charted with vertical bars. Note the insignificant variation in power yet substantial difference in temperature.

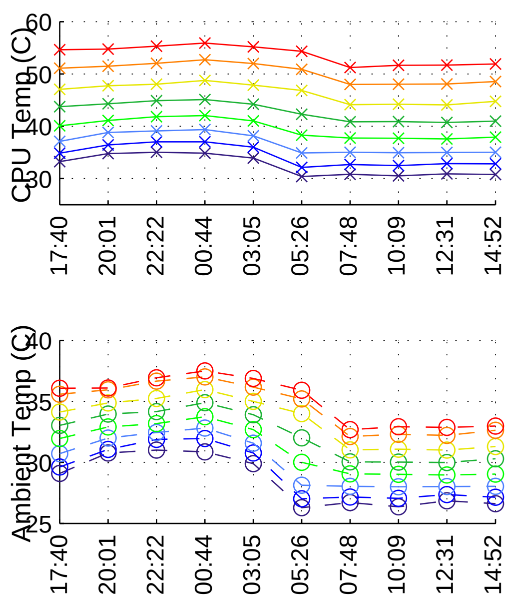


Figure 18: CPU and ambient temperatures recorded while `daxpy` repeated 10x at each p-state. External influences caused the ambient temperature to vary over a range of 5 degrees, reflected in CPU temperature variation.

4 SPEC CPU2000 Suite

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. The official SPEC rating method would first perform an initialization routine for all benchmarks, then launch each benchmark in succession. For logistical reasons, we split the suite into individual benchmarks, which produces a short period between each benchmark invocation in which data are not recorded. We use each benchmark's trace files separately, and also concatenate all benchmark data in execution order to reconstruct the full suite.

4.1 Individual Benchmarks

Figures 19 through 22 show in detail the effect of p-states on CPU temperature and power for two benchmarks, `mcf` and `galgel`. Each benchmark executes with every DVFS p-state, held constant for the duration of the benchmark. `Mcf` is memory bound, and even at the highest frequency p-state consumes less than 12 Watts on average. Intermittent power spikes are apparent in each p-state, with greater magnitude for higher frequencies. The CPU temperatures are fairly steady for each p-state, and range from about 34 to 46 degrees C. The power spikes seem to have little effect on the temperature, with thermal fluctuations of 1 C, imperceptible from the rounding error of the thermal sensor, for most cases.

`Galgel` consumes about 16 Watts at the highest frequency, down to about 3 Watts at the lowest-frequency p-state. `Galgel` exhibits periodic high-to-low power swings during a portion of the benchmark, with a distinctive zig-zag power pattern at higher frequencies. At lower frequencies, the core processor stalls for fewer cycles at lower frequencies (the memory speeds are unchanged and lowering the core speed provides a better match between core and memory), attenuating the bursty behavior observed at higher frequencies. As a result, the zig-zag power pattern is less noticeable at the low end of the frequency range. The temperature recorded for `galgel` reflects the power trends. The temperatures for high-frequency p-states are higher than for `mcf` due to the higher heat output from higher power consumption, and the temperature fluctuations are greater at higher frequencies during the zig-zag power periods than during that application phase at lower frequencies. The temperatures range from 32 degrees C to 56 degrees C, a larger range than for `mcf`.

4.2 Full Suite

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. The official SPEC rating method would first perform an initialization routine for all benchmarks, then launch each benchmark in succession. For logistical reasons, we split the suite into individual benchmarks, which produces a short period between each benchmark invocations in which data are not recorded. We concatenate each benchmark's data in execution order to reconstruct the full suite. Figure 23 shows moving averages over 1 second for each CPU temperature, ambient temperature, and CPU power for the SPEC benchmark suite executing at a fixed p-state of 2 GHz. In the figure, note that ambient power ramps to a stable region within about ten minutes, while the CPU temperature fluctuates continuously within a 5-10 degree range in response to power dissipation.

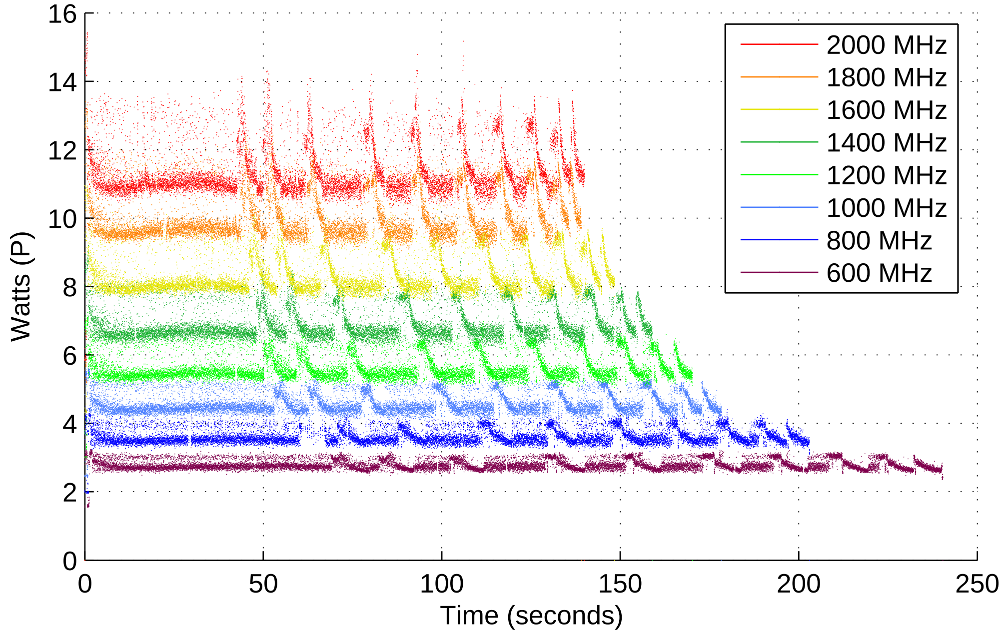


Figure 19: CPU power for `mcf` benchmark at each DVFS p-state.

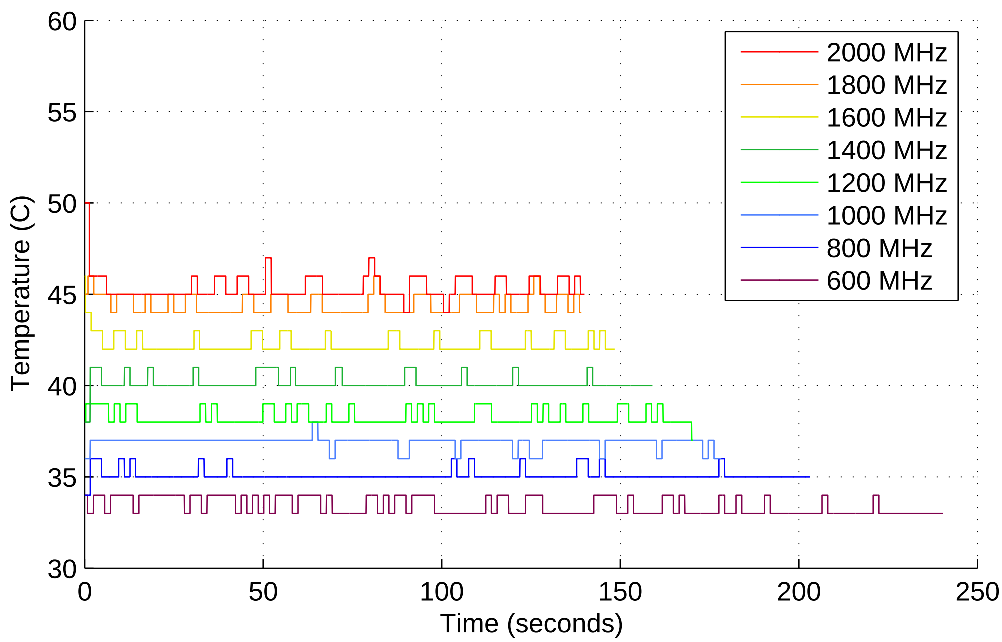


Figure 20: CPU temperature for `mcf` benchmark at each DVFS p-state.

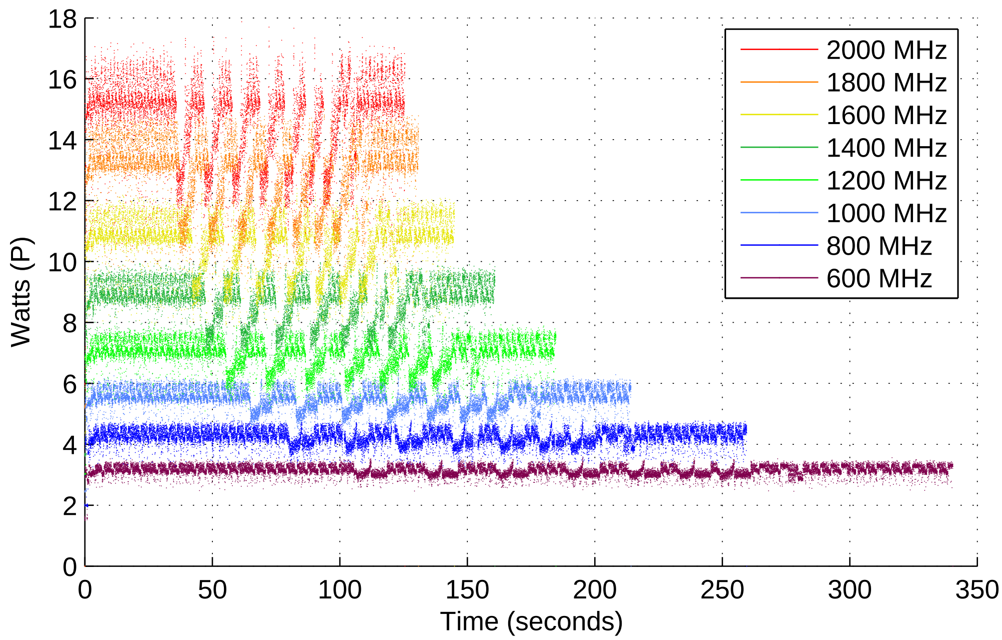


Figure 21: CPU power for `galgel` benchmark at each DVFS p-state.

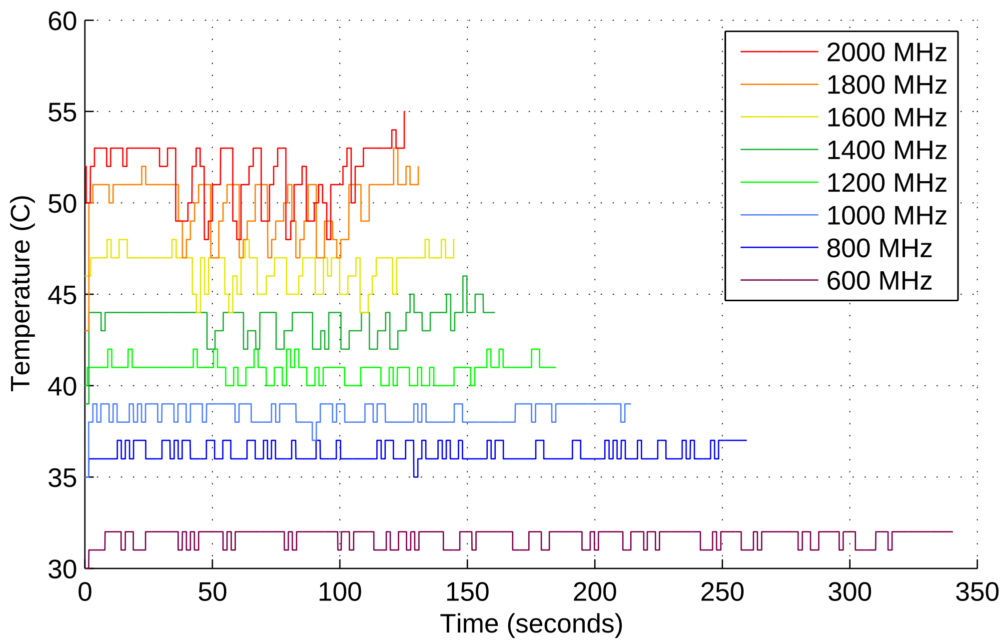


Figure 22: CPU temperature for `galgel` benchmark at each DVFS p-state.

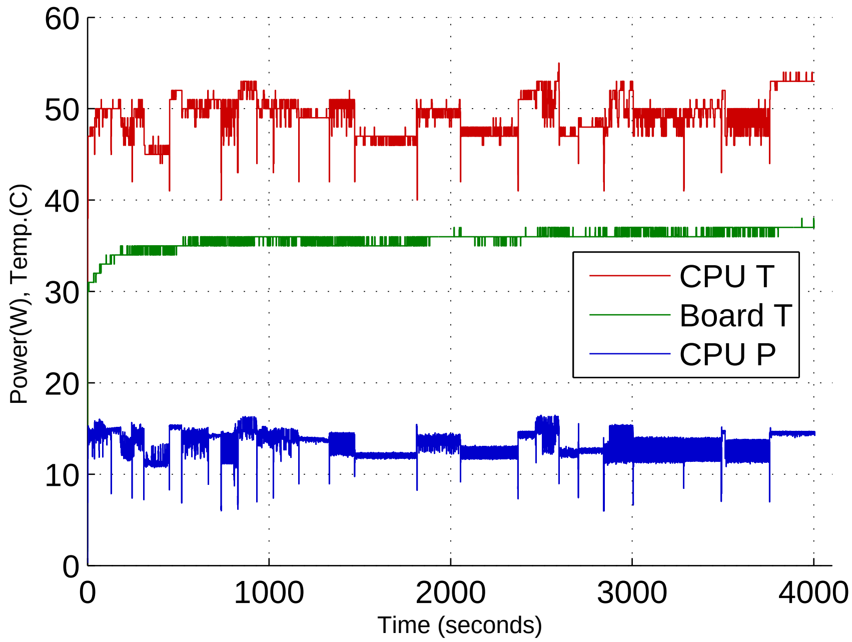


Figure 23: SPEC CPU2000 suite executing at 2GHz: 1-second moving average power and temperatures.

Figure 25 shows the effect of p-state on CPU temperature for the SPEC benchmarks. The full SPEC suite executed with each of the eight p-states while the fan spun at a high speed, approximately 4500 rpm. The graph plots the mean CPU temperature for each benchmark, ordered left to right from `gzip` through `apsi`, in execution order. Also plotted are the minimum and maximum recorded CPU temperatures as vertical error bars. Similar to the microbenchmark behavior, we observe that temperature is less dependent upon power in the low-power range, and more dependent on power in the high-power range. Note the similar power among benchmarks, yet fluctuating temperature readings, at 600 MHz. At mid-range and higher power levels, the shapes of the thermal curves more closely match the power curves. Temperature variation is larger for higher frequencies than lower frequencies, with greater minimum-maximum ranges and also larger differences between benchmark mean temperatures. It is evident that the interaction between workload characteristics and p-state influences power and CPU temperature. For example, the benchmark `mcf` at the 2000 MHz p-state exhibits a mean temperature similar to `crafty` at 1600 MHz.

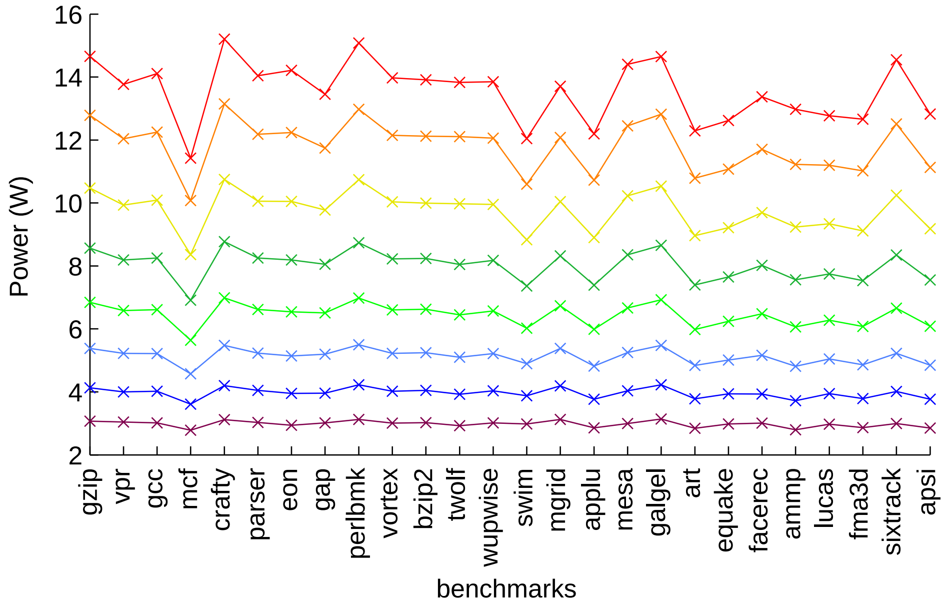


Figure 24: Mean power for each SPEC CPU2000 benchmark, at each DVFS p-state: 2GHz (red) down to 600 MHz (purple).

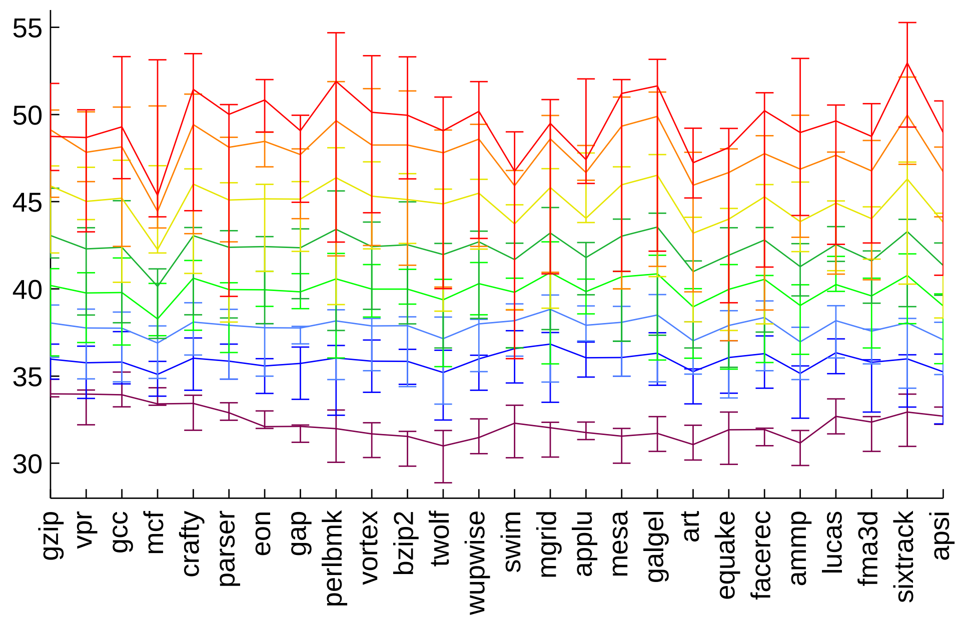


Figure 25: Mean CPU temperature for SPEC benchmarks at each DVFS p-state: 2GHz (red) down to 600 MHz (purple). Minimum and maximum temperatures indicated with vertical error bars.

5 Conclusion

Dynamic power and thermal management has become an important consideration as computing systems throughout the spectrum, from mobile devices to densely packed server racks, face serious power-related issues. System solutions to power and thermal management vary to some degree; for example, server rooms require industrial-strength air conditioning, while notebook computers must operate with a small and reasonably quiet fan, yet microprocessors in diverse settings face common problems of current supply, power density, and heat dissipation. In this study, we characterized the power and thermal response of a Pentium M system to answer the following questions:

- What are the timescale and magnitude of thermal response to p-state changes?

  We observe that the CPU temperature initially changes quickly, within 50 ms, in response to stepped power consumption, and the ambient temperature responds more slowly, over a period of minutes.

- What is the relationship between power and CPU temperature?

  The relationship between power and temperature is complex. We observed under tightly controlled conditions that with sufficient cooling capacity and steady ambient temperature and workload activity, CPU temperature is a linear function of power consumption. However, temperature depends on the interaction between cooling capacity and power dissipated. We induced high temperatures by disabling the CPU fan and executing a high-activity workload at a high frequency. We found that power and temperature form a 'thermal runaway' feedback loop where power increases, raising the temperature, which in turn increases the power, etc., only for high-power combinations of p-states and workloads. Temperature for low-power cases did not exhibit thermal runaway even when the CPU's heatsink fan was turned off.

  The CPU temperature depends upon the ambient temperature, as well as the power dissipation. A change of approximately 5 degrees in the ambient caused a corresponding 5-degree difference in the measured CPU temperature, regardless of the power levels.

- Can DVFS control CPU temperature?

  DVFS does have a significant impact on CPU temperature; however, fan speed, ambient temperature, and workload also influence temperature. Used in conjunction with run-time environment information, such as current temperature, power, and/or workload behavior, DVFS could be used to manage CPU temperature.

The measured data from live hardware illustrate the complex interaction between CPU and ambient temperatures and provide insight by identifying the magnitude and timescale of responses to power and thermal management control. Future work will include applying the effects of temperature on power, as well as power on temperature, on dynamic power and thermal control.

6 Acknowlegements

We would like to thank the Austin Research Lab for their generous support and use of the instrumented Pentium M system. Special thanks to Karthick Rajamani and Juan Rubio for developing the infrastructure and also to Soraya Ghiasi and Freeman Rawson and Tom Keller for the opportunity to collaborate with the ARL power-aware systems team.

References

[1] H.-P. Corporation, I. C. M. Corporation, P. T. Ltd., and T. Corporation. Advanced configuration and power interface specification, revision 3.0b. http://www.acpi.info, October 2006.

[2] H. Hanson and S. W. Keckler. Power and performance optimization: A case study with the Pentium M processor. In *Proceedings of the 7th Annual ACAS Conference*, 2006.

[3] Intel. Pentium M Processor on 90 nm Process with 2-MB L2 Cache Datasheet, January 2005. http://www.intel.com/design/ mobile/ datashts/ 302189.htm.

[4] Intel Corp. Enhanced Intel SpeedStep technology. http://support.intel.com/support/processors/mobile/pm/sb/CS-007981.htm, Jan. 2006.

[5] C. Lefurgy, K. Rajamani, F. Rawson, W. Felter, M. Kistler, and T. W. Keller. Energy management for commercial servers. *Computer*, 36(12):39–48, 2003.

[6] Radisys Corporation. Endura LS855 Product Data Sheet, October 2004. http://www.radisys.com/oem_products/ ds-page.cfm? productdatasheetsid=1158.

[7] 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.