Application-Specific Energy Modeling of Multi-Core Processors

Getov, Vladimir, Macduff, M., Kerbyson, D.J. and Hoisie, A.

Application-Specific Energy Modeling of Multi-Core Processors[1]

Vladimir GETOV[a,2], Matt MACDUFF[b], Darren J. KERBYSON[b] and Adolfy HOISIE[b]
[a] *University of Westminster, London, UK*
[b] *Pacific Northwest National Laboratory, Richland, WA, USA*

Abstract. Recent developments of high-end processors recognize energy monitoring and tuning as one of the main challenges towards achieving higher performance given the growing power and temperature constraints. Our thermal energy model is based on application-specific parameters such as consumed power, execution time, and equilibrium temperature as well as hardware-specific parameters such as half time for thermal rise or fall. As observed with the out-of-band instrumentation and monitoring infrastructure on our experimental cluster with air cooling, the temperature changes follow a relatively slow capacitor-style charge-discharge process. Therefore, we use the lumped thermal model that initiates an exponential process whenever there is a change in processor's power consumption. Experiments with two codes – Firestarter and Nekbone – validate our approach and demonstrate its use for analyzing and potentially improving the application-specific balance between temperature, power, and performance.

Keywords. Thermal energy model, application-specific characterization, power and performance workload analysis, out-of-band instrumentation and monitoring

1. Introduction

In recent years, the development of new concepts and implementations for power and energy instrumentation and analysis of modern scalable computers has been the primary concern for building exascale supercomputers in the near future [1]. Subsequently, thermal studies and analysis have gradually been included in this set of research and development challenges. The major distinguishing factor between main stream thermal energy efficiency efforts and the priorities of the high-performance computing domain is the never-ending motivation for achieving higher performance.

Thermal science and engineering is a very well developed and understood discipline [2]. The main processes that facilitate the transfer of thermal energy are conduction, convection, and radiation with one or a combination of them participating in a heat transfer process. If we consider a solid body that is heated by electrical current, its temperature will keep increasing until the rate of heat generated by the electrical current balances the aggregate rate of heat loss for this body. This allows a rough estimate of the transient temperature rise by electrical current and the transient temperature fall that follows when the electrical current is no longer applied. In the

[1] This is a modified and extended version of a previously published paper [3].

[2] Corresponding Author, Distributed and Intelligent Systems Research Group, University of Westminster, 115 New Cavendish Street, London W1W 6UW, UK; E-mail: V.S.Getov@westminster.ac.uk.

usual case, the heat transfer depends on the material and geometry of the body as well as on the surroundings. Models and corresponding equations for this usual case have been studied and developed in recent years and are now part of the modern thermal engineering. The lumped thermal model in particular has been recognized as a very good approximation of thermal processes in low-voltage microelectronic devices including processors and other kinds of semiconductor components [4].

The electrical analogy of the lumped thermal model is based on an RC circuit that models the conduction in a solid body with another resistance in parallel to the capacitance to model the convection heat transfer with the surroundings. As a criterion for good approximation, lumped systems analysis use the Biot number which is the ratio of the internal resistance (conduction) to the external resistance to heat convection. Since lumped capacitance systems assume that there is no spatial variation of temperature, the internal resistance for the ideal case is zero and the Biot number will be zero too. It is generally accepted that the lumped system analysis is applicable if the Biot number is smaller than 0.1. In other words, for lumped system modeling the external convection resistance should be at least an order of magnitude larger than the internal conduction resistance. Therefore, relatively small bodies with high thermal conductivity are good candidates for the lumped thermal capacitance model where the conduction resistance is negligible and does not participate in the equations.

In our work, presented in this paper, we have considered a system including two lumps – each having their own and different thermal capacitance. The system temperature as measured by a thermal sensor keeps changing only in time while being the same everywhere in the two lumps. Using the PowerInsight out-of-band infrastructure on our SeaPearl cluster, we have observed that those temperature changes follow a capacitor-style charge-discharge process. Therefore, we use the lumped thermal capacitance model with all associated assumptions and simplifications. This proves to be an accurate approximation providing excellent fit between measured and calculated temperatures for specific applications.

The rest of this paper is structured as follows. Section 2 introduces our thermal energy model and the two core parameters – the asymptotic equilibrium temperature and the half time for thermal rise or fall. Section 3 provides an overview of the experimental setup while Section 4 presents thermal, power and performance results with corresponding discussions. Finally, Section 5 concludes the paper.

2. Thermal Model of Processors with Air Cooling

2.1. Model Design

The fundamental modes of heat transfer are conduction, convection, and radiation [2]. Heat convection is not present in solid bodies, since neither bulk current flows nor significant diffusion can take place in solids. Normally, conduction is the major internal mode of heat transfer in solid bodies, while heat dissipation in the surroundings (usually air) is done by convection. Since our system involves two solid lumps with no internal convection and temperatures well below 100 degC which indicates lack of any radiation, the internal heat transfer that takes place is by conduction only. Also, part of the heat generated by the electrical current flow is stored in the two lumps. Therefore, in our case the generated heat equals the sum of the stored heat, the heat transferred by internal conduction and the heat exchanged by convection with the surroundings.

The generic thermal energy model, when the heat transfer depends on the material and geometry of the body as well as on the surroundings, is based on the following main assumptions:

- There exists a thermal equilibrium when the heat rate generated by the electrical current is in balance with the aggregate rate of heat expelled by the body.
- The heat transfer coefficients for both convection and conduction can be treated as a lump constant, α. Strictly speaking, this parameter is not exactly constant but the error associated with this assumption is negligibly small.
- Depending on the packaging and air flow rate, it is expected that the thermal properties and constraints of the heat sink and the processor package are different [5]. Therefore, we build our model by applying superimposition of those two lumps and the corresponding equations.
- The model is relatively simple with only two lumps – the processor package and a basic heat sink without a local fan mounted on top of the package with a temperature sensor inserted right in the middle between the two (see Figure 1).

The idle equilibrium temperature in a quiescent state (no application software workload) measured by the thermal sensor between the processor package and the heat sink is only around 3 degC higher than the ambient temperature. Therefore, it can be assumed that the idle and the ambient temperatures are the same. Another way to approach this small difference is to express the idle temperature as the equilibrium achieved above the ambient (power-off) temperature when the chip's electrical consumption is zero at the power-off state. This assumption simplifies and clarifies further the model while we obtain a very good fit of the experimental results.

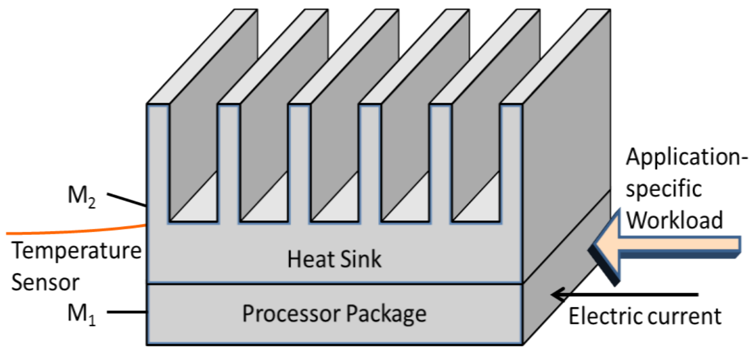

Figure 1. Processor package with a heat sink and a temperature sensor.

2.2. Notation

The following variables and parameters are essential for the equations of simple transient thermal processes in a solid body.

P – electric power; $P = I \text{ x } U = I^2R$;

I – electric current; R – Ohmic resistance;

$T(t)$ – transient temperature as a function of time only;

T_0 – temperature at the start of a transient thermal process;

T_∞ – asymptotic temperature at equilibrium achievable after infinitely long time.

Thus, the asymptotic equilibrium temperature is a function of the consumed electric power by the processor package and the body parameters:

$T_\infty = I^2R/\alpha A$;

$M = cW/\alpha A$ – is a time "constant" given the assumptions discussed in Subsection 2.1 are satisfied. It depends on the material and the geometry of each lump combining all physical parameters of the solid body and depends on the following parameters:

c – specific heat of the material treated as a constant in the temperature range 0-100 degC;

W – body weight and A – surface area;

α – a lumped constant, assuming that the heat transfer coefficients for both convection and conduction can be treated as a constant;

cW – describes the stored heat by a given body/lump;

αA – describes the heat loss by conduction for a specific body/lump.

2.3. Heating and Cooling Equations

During a differential time interval Δt the temperature of a solid body rises or falls by a differential amount ΔT. The energy balance of the solid for the time interval Δt follows the First Law of Thermodynamics which states that the heat transfer into/from the solid body during Δt equals the increase/decrease in the energy of the body during the same interval. Let us assume that at the start of the process the time is t_0 and the initial temperature is T_0. Then, we let the transient thermal process continue for infinitely long time period. In this case, the generic equation is:

$$\frac{T(t) - T_\infty}{T_0 - T_\infty} = e^{-\frac{t}{M}} \tag{1}$$

The lumped system analysis and the above equation are equally valid for both heating and cooling with the transient temperature following a capacitor-style charge or discharge processes. The exponential equations for the temperature rise or fall relative to the current temperature of the solid body at the start of the process in those two cases are derived in the following sub-sections.

2.3.1. Heating Equation

If the temperature of the next equilibrium is higher than the current temperature of the solid body, the system starts a heating transient process, where:

$\tau^h(t) = T^h(t) - T_0;\ \tau^h_\infty = T^h_\infty - T_0;$

After substituting in the generic equation (1) above and some rearrangements, the exponential equation for heating is:

$$\tau^h(t) = \tau^h_\infty\left(1 - e^{-\frac{t}{M}}\right) \tag{2}$$

2.3.2. Cooling Equation

If the temperature of the next equilibrium is lower than the current temperature of the solid body, the system starts a cooling transient process, where:

$\tau^c(t) = T^c(t) - T^c_\infty;\ \tau^c_0 = T_0 - T^c_\infty;$

After substituting in the generic equation above and some rearrangements, the exponential equation for cooling is:

$$\tau^c(t) = \tau^c_0 e^{-\frac{t}{M}} \tag{3}$$

2.4. Asymptotic Equilibrium Temperature

At the beginning of a transient thermal process, the solid body could be in a steady state (equilibrium temperature) or in the middle of another thermal process that had started earlier (transient temperature). This new process can be either for heating or for cooling and it is therefore important to find out if the temperature is going to rise or to fall because the exponential equations (2,3) for the two processes are different. The starting temperature, T_0, which determines the initial condition for the two equations, is a component for both the transient temperature and for to the aggregate asymptotic temperature at equilibrium, T_∞, achievable after infinitely long time.

Unfortunately, in real systems T_0 is often unstable and although its variations are usually small – within the range of a few degrees only – it is better for clarity and accuracy to exclude T_0 from the evaluation of the asymptotic temperature at equilibrium. Therefore, it is preferable to use τ_∞ in the asymptotic analysis of the heating and cooling processes.

For heating, the formula for the asymptotic equilibrium temperature, τ^h_∞, is:

$$\tau^h_\infty = \frac{\tau^h(t)}{1 - e^{-\frac{t}{M}}} \tag{4}$$

Since $e^{-\frac{t}{M}}$ is zero for infinitely long transient process time, it is obvious from equation (3) that for cooling $\tau^c_\infty = 0$.

When building our lumped thermal model, we apply superimposition of two lumps with the corresponding equations. Let us consider the superimposition of the exponential parts of those equations without taking into account the temperature at the start of the process, T_0, which can always be added later.

$$\tau(t) = \tau_1(t) + \tau_2(t) \tag{5}$$

After substituting with equation (2) for heating, we obtain:

$$\tau^h(t) = \tau^h_{1\infty}\left(1 - e^{-\frac{t}{M_1}}\right) + \tau^h_{2\infty}\left(1 - e^{-\frac{t}{M_2}}\right) \tag{6}$$

Assuming M_1 and M_2 are known, one can now use the heating equilibrium equation (4) to evaluate separately $\tau_{1\infty}$ and $\tau_{2\infty}$:

$$\tau^h_\infty = \tau^h_{1\infty} + \tau^h_{2\infty} \tag{7}$$

Finally, the aggregate asymptotic equilibrium temperature at the end of a heating process is simply:

$$T^h_\infty = \tau^h_\infty + T_0 \tag{8}$$

Let us review again the very beginning of the execution of an application code on a processor package when it is in a quiescent state. In this case, T_0 is the idle equilibrium temperature (no application software workload yet), while $\tau^h{}_\infty$ depends solely on the application workload via the additional consumed power during the execution. Therefore, the aggregate asymptotic equilibrium temperature includes three components with clear physical interpretation for each of them as follows:

1. Two of those components are static – the ambient (power-off) temperature and the idle equilibrium temperature in a quiescent state which is usually a few degrees higher because of the system software activity while the processor is idle.
2. However, the third component – the asymptotic equilibrium temperature $\tau^h{}_\infty$ – is transient. It is application-specific and characterizes the workload of the application code.

If we consider again equation (5), but this time for cooling, and take into account that $e^{-\frac{t}{M}}$ is zero for infinitely long process time, the aggregate asymptotic equilibrium temperature for cooling is:

$$T^c_\infty = T_0 - \tau^c_0 \tag{9}$$

2.5. Half Time for Thermal Rise or Fall

Let us consider further the two transient thermal processes – heating for temperature rise (increasing concave down curve) and cooling for temperature fall (decreasing concave up curve). In the heating exponential equation (2), $\tau^h{}_\infty$ is the asymptotic equilibrium temperature reachable after infinitely long time. It is practically much more useful to solve this equation for half the asymptotic temperature:

$$\frac{\tau^h_\infty}{2} = \tau^h_\infty\left(1 - e^{-\frac{t}{M}}\right) \tag{10}$$

which turns into a simple formula:

$$e^{-\frac{t}{M}} = \frac{1}{2} \tag{11}$$

If we take the natural logarithm of this equation the result is

$$t_{\frac{1}{2}} = Mln2 \tag{12}$$

This shows that for a specific solid body the time for half thermal rise depends only on M. Similar to the above derivation for heating, one can solve the cooling exponential equation (3) for half the asymptotic temperature which has the same solution (12) as the one for a heating process.

$$\frac{\tau_\infty^c}{2} = \tau_0^c e^{-\frac{t}{M}} \tag{13}$$

This proves that for a specific solid body the half time for thermal fall again depends only on M and it is the same as the one for thermal rise.

The above leads to the following important conclusions:

1. The half time for thermal rise and fall is the same for either heating or cooling and depends only on the constant parameter M.
2. The half time for thermal rise and fall characterizes the thermal properties of the hardware and can be used for comparisons between different constructions involving different processors.

3. Experimental Setup

3.1. Runtime Instrumentation

In order to explore and validate the thermal model on current compute systems, we require instrumentation with sufficient resolution in a realistic environment and appropriate applications that provide a consistent workload.

3.1.1. Background

The concept of runtime instrumentation and monitoring is not new but has been attracting increasing attention in recent years because of the very high level of complexity of current processors. This can be a valuable resource as it allows runtime monitoring with potential for dynamic tuning depending on pre-set criteria. In considering the runtime instrumentation for the purposes of our project, it was important to take into account several dimensions of monitoring including:

- Sampling frequency;
- Measurement overhead;
- Locality/positioning of sensors**.**

Specifically, we needed to measure the consumed power and temperature of individual processors for jobs that may last between a few seconds and 10-15 minutes, without adding extra workload to the CPU.

3.1.2. Tiers of Monitoring

We define several tiers of instrumentation and monitoring as follows:

1. External sensor that observes a system as a whole. This might include a power strip sensor or a 'Wattsup' power meter [6]. For temperature, this could be a thermometer placed in the back of a computer rack. This is not invasive to a system and, in many cases, is already available in many environments. It does not provide the locality of isolating a single CPU socket and is often limited in its sampling frequency.

2. In-band monitoring based on built-in sensors can provide very specific locality. Features like RAPL (running average power limit) provide direct access to MSRs (model specific registers) located in the processor to model power usage [7]. It is necessarily in-band, adding additional load to the system. These capabilities are hardware and OS dependent but they have become very common in modern computers. They can also provide very high sampling rates for precise measurements.
3. Out-of-band power and temperature instrumentation and monitoring implemented by external infrastructure can provide locality without impacting the system being measured. Many current processors commonly include BMC (baseboard management controller), which can provide access to several sensors that are used by the system to monitor system health. However, the BMC sampling frequency is typically slow (less than 1 Hz).

3.1.3. PowerInsight – Overview

In order to provide the required locality, performance and the advantages of out-of-band collection we considered a unique tier 3 solution by Penguin. The PowerInsight 2.1 instrumentation and monitoring system [8] meets all the requirements to validate our model. As shown in Figure 2, this infrastructure allows for multiple thermal and power sensors to be placed within different compute systems providing measurements of individual sockets, memory and other components without introducing any power or performance overhead while running an application code.

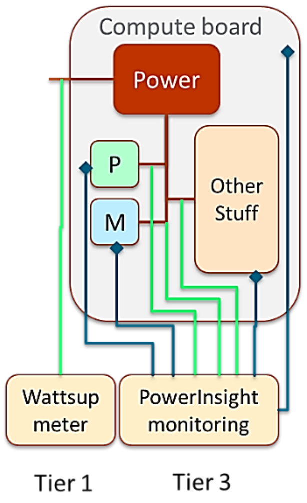

Figure 2. Instrumentation block diagram.

By using a dedicated BeagleBone data collection system (see Figure 3) embedded in each computer, data can be collected at rates up to 1 KHz with no impact on the compute workload. This provides exceptional resolution of consumed energy on the individual sockets we are studying with close coupling of thermal and power readings. Because of the adopted design approach, the PowerInsight infrastructure is embedded in normal computer servers that operate in a typical computer room environment.

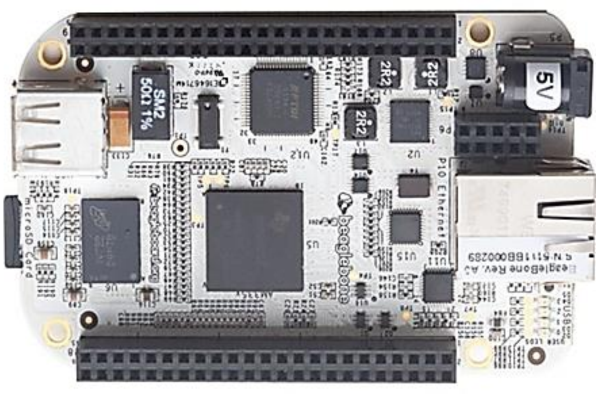

Figure 3. BeagleBone embedded data collection system.

3.2. Test Processors

Our test system – the SeaPearl cluster – was part of a large computing center providing a climate controlled environment with air cooling. The experimental setup was based on the PowerInsight infrastructure with sensors to measure the total power of each socket and the temperature directly between the heat sink and the processor. Using the PowerInsight instrumentation for our experiments, we collected data at approximately 5Hz for both power (Watts) and temperature (degC). The three processors that were studied and used to validate our thermal energy abstraction and model have recent Intel CPUs and 128 GB of memory with some more details included in Table 1. They all provide homogeneous compute environment with differences in the number of cores and other key parameters while using the same 22 nm technology [9, 10]. Each progressive processor model introduces capabilities that may affect power and thermal budget.

Table 1. Details of the three Intel processors.

Details	**IvyBridge E5-2680 v2**	**Haswell E5-2630 v3**	**Haswell E5-2698 v3**
No of cores	10	8	16
Cache size	25 MB	20 MB	40 MB
Base frequency	2.8 GHz	2.4 GHz	2.2 GHz

The generic processor architecture for homogeneous multi-core Intel server processors which covers the three models selected for our experiments is depicted on Figure 4. The cores provide dedicated resources for the execution of one or two threads if hyper-threading is enabled. In contrast, uncore components are shared between all threads and the workload balance between the cores and the uncore depends on the number of active cores. Further analysis of the internal workload distribution can follow the block-diagram shown in Figure 4 unless more specific architecture details are needed.

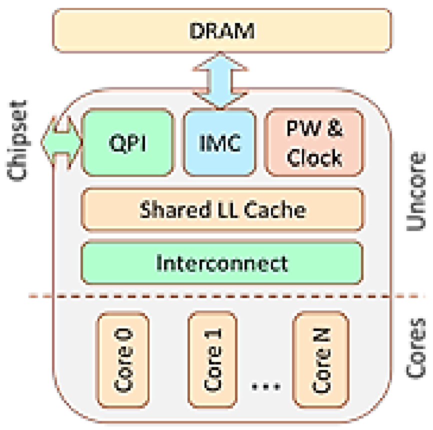

Figure 4. Generic processor architecture.

3.3. Application Use Cases

Computer workload is defined mainly by the currently running application and depends on both the code and the data. In general, this application-specific workload can be characterized by its performance profile and by its energy profile. Usually, an important target for the application developers is to reduce as much as possible the time-to-solution which leads to higher performance and lowers the energy budget for a particular code. In order to generate workloads on the system we utilized two different applications.

3.3.1. Firestarter

The Firestarter software is designed to stress a processor to its fullest [11]. By combining with CPU affinity, one can selectively load certain numbers of cores on a given socket with this 'stress code' utility. The basic energy profile of Firestarter has negligible startup and shutdown artifacts, providing very high electric power workload in the form of discrete and consistently uniform time steps.

3.3.2. Nekbone

The Nekbone suite of benchmarks is a well-established set of thermal hydraulics mini-application tests that also provide a measure of work [12]. This enables us to consider not just temperature and power but also performance. In our experiments, we have used the Nekbone multigrid preconditioner benchmark.

4. Results and Model Validation

Experiments have been conducted in a datacenter conforming to environmental class A1 according to the ASHRAE (American Society of Heating, Refrigerating and Air-Conditioning Engineers) requirements with tightly controlled ambient temperature within the range 18–27 degC (64–81 degF) [13]. In this environmental class, the ambient temperature is normally lower than the idle temperature of the chip which makes sense and is consistent on all processors.

As demonstrated by our own measurements the ambient temperature in the surroundings area throughout the experiments did not change more than 0.2–0.3 degC within the range 18.9–23.3 degC depending predominantly on the position (high or low) in the cabinet. This also confirms that the air temperature is very well controlled inside the server cabinets of our SeaPearl cluster.

4.1. Thermal Model Validation

The experiments include temperature and power consumption measurements on idle processors as a starting point and then gradually incrementing the number of active cores using explicit one-to-one assignment of threads to cores. We fit the thermal experimental data (120 seconds runtime) for the maximum workload on all three processors by applying the equations derived in Section 2 as shown in Figures 5.a, 5.b and 5.c. Table 2 summarizes the main parameters derived from the measurement data.

Table 2. Measured main parameters of the three processors.

Parameters	**IvyBridge E5-2680 v2**	**Haswell E5-2630 v3**	**Haswell E5-2698 v3**
Half time	20.6 sec	26.9 sec	29.1 sec
Equilibrium temperature	61.7 degC	61.2 degC	72.0 degC
Load power	128 W	96 W	156 W
Average Std. Dev.	0.20 degC	0.22 degC	0.17 degC

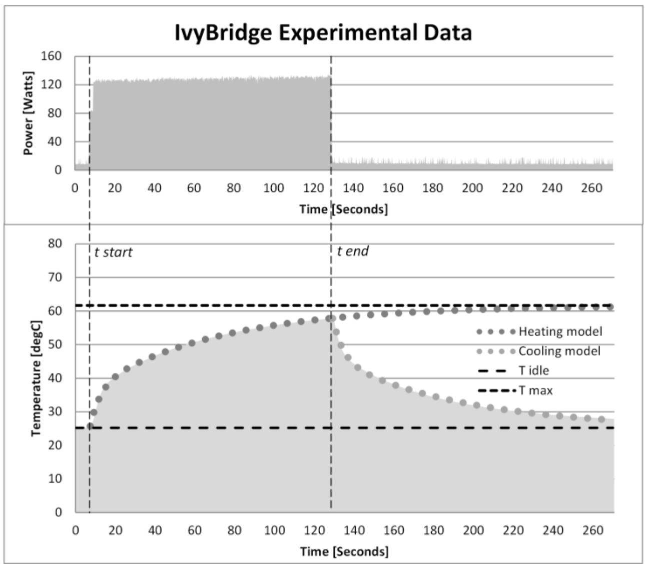

Figure 5.a. Thermal model validation for the 10-core IvyBridge processor.

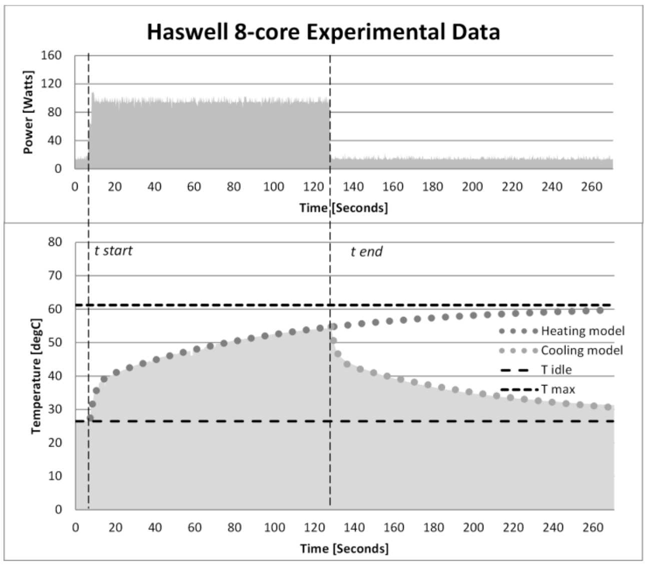

Figure 5.b. Thermal model validation for the 8-core Haswell processor.

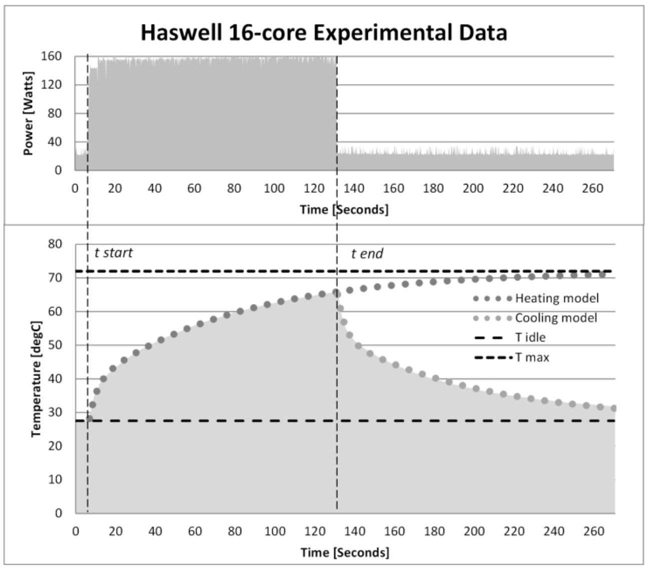

Figure 5.c. Thermal model validation for the 16-core Haswell processor.

4.2. Temperature vs Consumed Power

While validating our thermal model for the three homogeneous current processors was certainly very important, our next step was to investigate the rise of the asymptotic equilibrium temperature when increasing the consumed power. In order to do that we have run experimental results measuring the temperature while increasing the number of active cores on a 10-core IvyBridge processor with a fixed workload per core and running time kept the same at 300 seconds (5 minutes). Since those experiments were longer, it was good to validate our model again by fitting the experimental results for 5 minutes as shown in Figure 6. This demonstrates again an excellent match between experimental data and exponential equation with the temperature getting much closer to the asymptotic equilibrium because of the longer runtime.

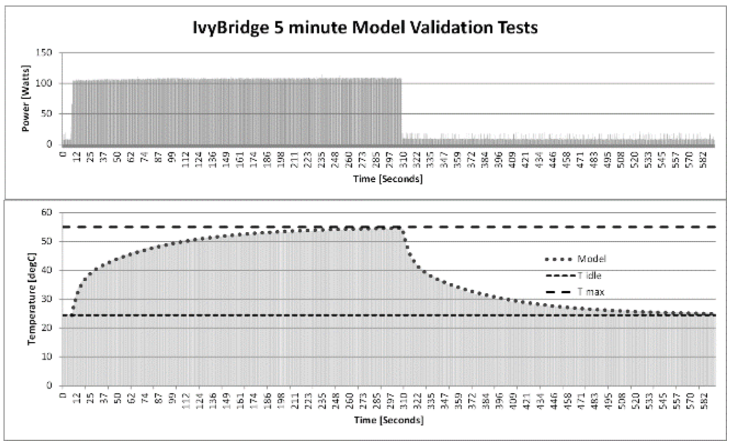

Figure 6. Thermal model validation for a 300-second run on the 10-core IvyBridge.

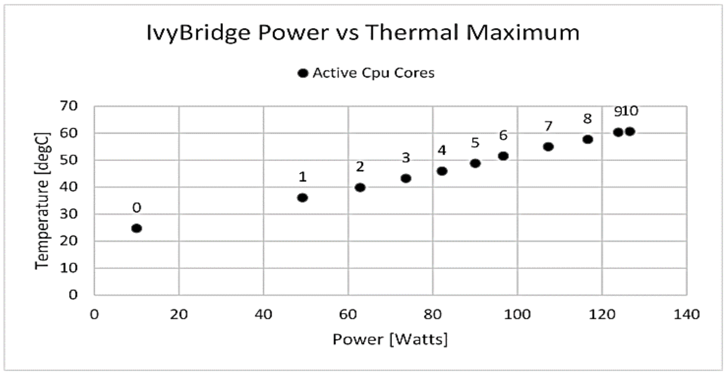

Figure 7. Comparison of consumed power and equilibrium temperature for different number of active cores.

Figure 7 shows how consumed power and equilibrium temperature scale when incrementing the number of active cores. The significant increase of the consumed power between zero and one active cores is clearly visible. This is due to the need to engage the uncore part of the chip even though only one core is active. This difference is also illustrated via the much larger distance along the x axes between zero and one active core. As the chip becomes fully loaded the increment in the consumed power between nine and ten cores becomes very small.

Table 3. Characterization parameters for different number of active cores on IvyBridge E5-2680v2.

Active cores	Load power	Equilibrium temperature	Model Std Dev	Half time heating	Half time cooling
1	49 W	36.1 degC	0.07 degC	20.4 sec	21.5 sec
2	63 W	39.9 degC	0.09 degC	22.7 sec	21.6 sec
3	74 W	43.2 degC	0.13 degC	22.4 sec	22.4 sec
4	82 W	45.9 degC	0.12 degC	22.2 sec	21.7 sec
5	90 W	48.8 degC	0.15 degC	22.8 sec	22.2 sec
6	96 W	51.5 degC	0.14 degC	22.1 sec	21.7 sec
7	107 W	55.0 degC	0.19 degC	20.0 sec	20.9 sec
8	116 W	57.7 degC	0.27 degC	20.5 sec	20.9 sec
9	124 W	60.3 degC	0.26 degC	21.0 sec	21.5 sec
10	126 W	60.6 degC	0.26 degC	21.6 sec	22.3 sec
Avg			0.17 degC	21.6 sec	21.7 sec

Table 3 lists the experimental results for both the consumed power and the derived equilibrium temperature for this set of experiments. In addition, we provide the derived half time values for both heating and cooling as well as the standard deviation which demonstrates an excellent fit between the raw data and the calculated values.

4.3. Performance Experiments

We have run the Nekbone code on a single 16-core Haswell processor package. This set of experiments has provided measurements for performance, consumed power, and temperature while incrementing the number of active cores and keeping the total Nekbone workload constant at 96 elements for the multigrid preconditioner benchmark as part of the suite.

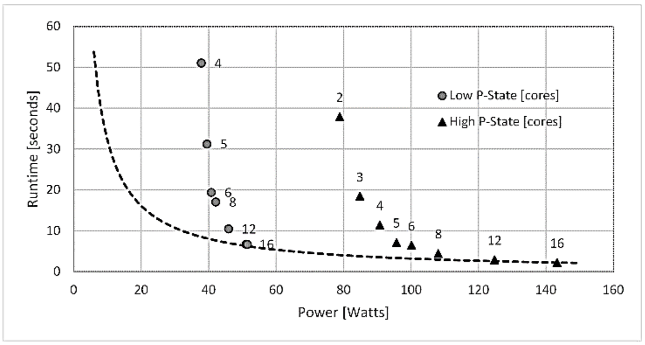

Figure 8. Power vs runtime for Nekbone on 16-core Haswell.

The Nekbone workload was very similar to the Firestarter results in Subsection 4.1 and Subsection 4.2. The power level would almost immediately step up to a constant load and then step back down to the idle power when it was complete. Thus, the capacitive heating and cooling response was very similar to our thermal model. Adding the third dimension of performance (runtime) provides an important connection of this work to real applications. To examine a variety of cases we not only varied the number of active cores but also the CPU scaling governor which controls the P-states of the

CPUs. We have focused on the two extreme cases – low P-state (*powersave* scaling governor) and high P-state (*ondemand* scaling governor).

In comparing the runtime against power produces a plot in Figure 8 that would be expected. As power increases with more cores, the performance also improves. A Pareto front is added for both high P-state and low P-state modes. The best solution is for the maximum number of active cores in high P-state mode. It is interesting to note that for this workload, there are other points that are very close to the optimum solution but at less power. In particular, the 16-core low P-state run uses about one third of the consumed power while staying very close to the Pareto front.

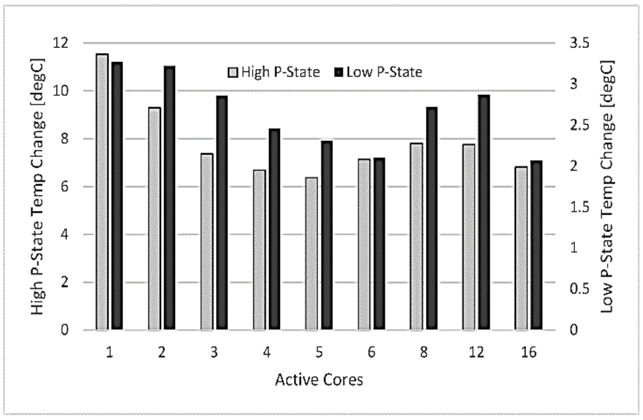

Figure 9. Temperature change on 16-core Haswell for different number of active cores.

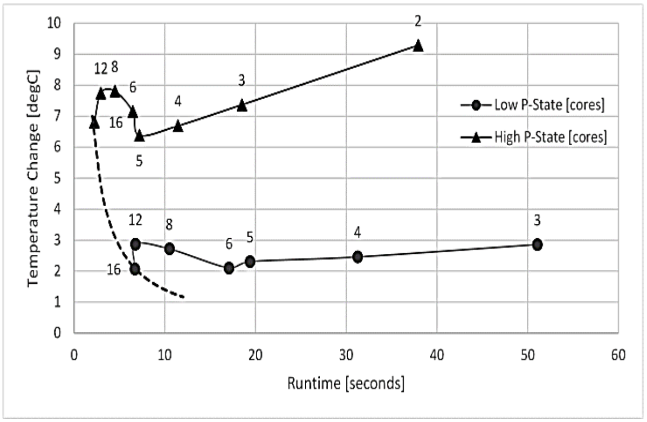

Figure 10. Runtime vs temperature change for Nekbone on 16-core Haswell.

Because of the varying runtime and the capacitive temperature response, the peak temperatures do not follow a straight forward trend. Instead, we see a valley in the temperatures (Figure 9) across the range of cores. This is also highlighted when we compare runtime with peak temperatures (Figure 10). The Pareto front on this figure still shows the optimal solution with 16 cores, although in this case it is the lower P-state that is the best. One can discern that as the runtimes get closer together with more cores, the increased consumed power must be driving the temperature faster than the relative decrease in runtime.

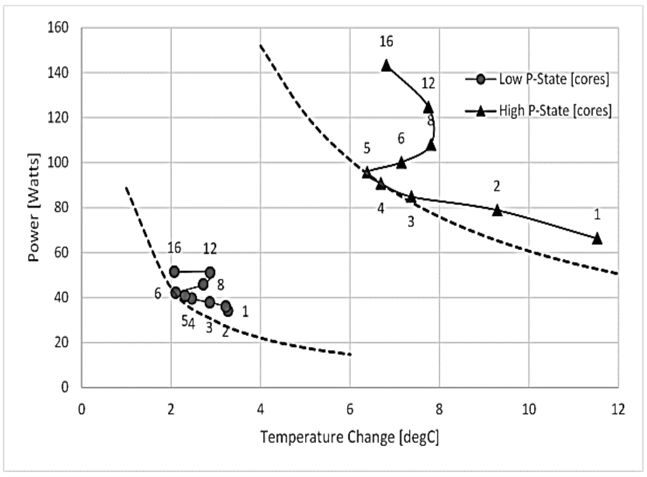

Figure 11. Temperature change vs power for Nekbone on 16-core Haswell.

However, as shown in Figure 11, the effect of this results in the Pareto minimum occurring with just a few cores running when we compare the consumed power and the asymptotic equilibrium temperature. This is an intriguing result to consider that workloads seeking to optimize power and temperature may find the best solution with the right mix of active cores. We also note that the lower P-state follows this same pattern but less pronounced because of the much-reduced range of power applied.

This creates an interesting area for further research into this effect across different applications and processors.

4.4. Cooling Management

In general, the main thermal management actions to be performed automatically by modern computer systems [13] when the working temperature keeps increasing are activated at runtime in the following order:

1. Cooling management – reliability threshold;
2. Performance and power management usually implemented via dynamic voltage and frequency scaling (DVFS) mechanisms – functional limits;
3. Shutdown – damage threshold.

These thermal management actions are normally activated automatically at runtime and would obviously affect the energy profile of the application code. Therefore, the energy profile of an application is likely to be different when running on different computer systems. We provide here only one initial experiment while further research in this area is part of our current and future work.

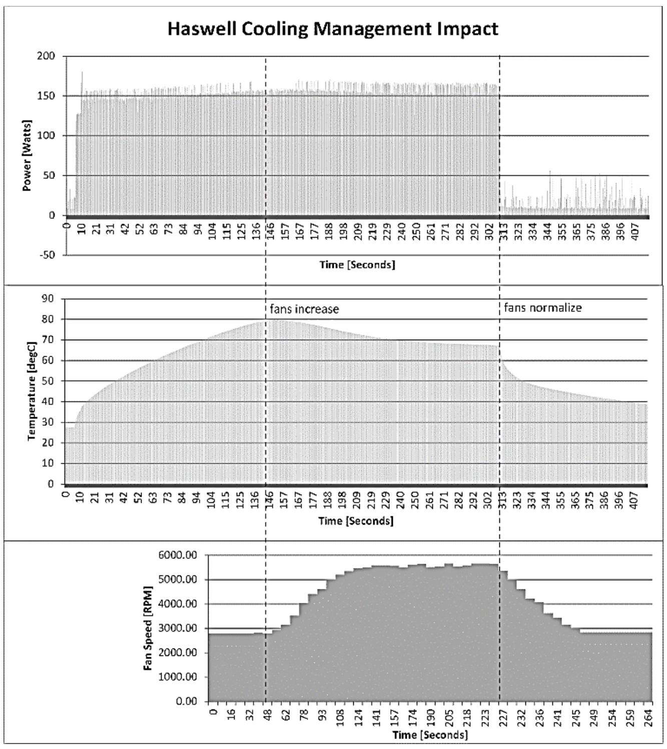

Figure 12. Cooling management impact on 16-core Haswell due to high temperature on socket 2.

We have conducted experiments measuring the consumed power and temperature while increasing the running time with the number of active cores and workload kept at the possible maximum using the Firestarter code. A significant part of this application-defined consumed power transforms into heat which could raise significantly the processor temperature. In some cases, the asymptotic equilibrium temperature as predicted by our model cannot be practically achieved because further increase to the execution time for heavy workloads leads to a temperature drop in the region around 80 degC for the second socket of our 16-core Haswell processor while the first socket is also fully loaded running the same code. This temperature drop happens because the automatic cooling management increases the speed of the four fans as shown on Figure 12. The consumed power by the two processors and the running of the code remain without any changes but the overall power consumed by the node increases by around 5% which amounts to ~25 Watts because of the higher speed of the four fans. At the same time the maximum temperature of the first socket remains much lower below 67 degC most likely because this processor, although fully loaded, is much closer to the four fans than the second one.

This experiment shows that usually the high temperature on the second socket is the most likely reason for activating the cooling management. Obviously, this depends on the construction of the node and its thermal design in particular but it requires more work and analysis of possible thermal differences between different sockets. In addition, the cooling management also depends on the workload balance between the two sockets which also depends on the application code. For example, lower workload on the first socket is expected to affect the temperature of the second socket and vice versa. Our current and future plans include further experiments and investigation of these issues.

4.5. Variable Energy Profile

Some of the application-specific parameters in our model such as power consumption and execution time have also been part of the so-called energy templates [14, 15]. Usually, the energy template for a given code is a series of time steps with constant consumed power and sharp changes between neighboring periods which could be different for different cores participating in the execution. In this more general case, our lumped thermal model is re-initiated whenever there is a change in processor's power consumption – normally at the beginning of the next time step. It then proceeds exponentially towards a new thermal equilibrium which is likely to be interrupted by the next time step in the energy template and so on.

As a demonstration of our approach towards analyzing real applications, we have generated an artificial variable workload as shown in Figure 13.

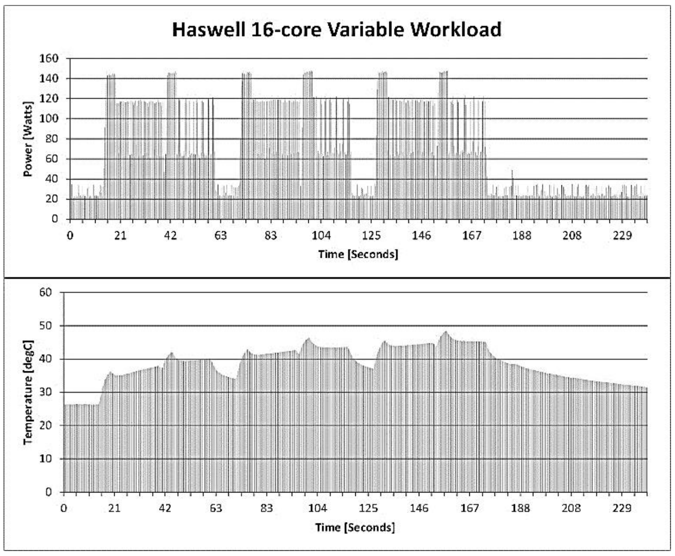

Figure 13. Consumed power and temperature for an artificially generated variable energy profile.

For a series of time steps with constant consumed power for each step the exponential equations from Section 2 can be substituted by two generic equations (heating and cooling) for the temperature change within a given time step based on the ending temperature of the previous step (equilibrium or not) and the next potential equilibrium which will be asymptotically reached if the time step is infinitely long. This approach can enable the development of a more realistic application-specific thermal modeling methodology incorporating energy templates which is part of our ongoing current and future research work.

5. Conclusions

In this article, we have presented our abstract thermal energy model as developed using the thermodynamics principles. The model is applicable to current processors and other similar electronic devices where the heat is generated by the transfer of electric into thermal energy depending on the application-defined workload. The exponential equations and the derivations are very similar to the ones used in several other areas of science and engineering such as nuclear physics and electrical engineering. The lumped thermal capacitance model and the two core parameters – asymptotic equilibrium temperature and half time for heating or cooling – have also been studied and used extensively although not particularly in computer science and engineering.

The novelty of our work is the interpretation of those two parameters which are the core of our model and characterize remarkably well the relationship between the thermal properties of the hardware design and the application-specific power workload generated by a given application code.

The asymptotic equilibrium temperature, in particular, is application-specific and characterizes the workload of the application code.

The half time for thermal rise and fall is the same for either heating or cooling and depends only on the constant parameter M. Therefore, in systems conforming to our model, the half time depends only on the parameters of the material but not, for example, on the differential temperature, ΔT, for a specific exponential thermal process. Hence, it characterizes the thermal properties of the hardware and can be used for comparisons between different constructions involving different processors.

In addition, our thermal energy model provides very accurate characterization with excellent fit between experimental data and calculated by the exponential equations values. The deviation between our model and the experimental results is less than 0.3 degC which corresponds to the uncertainty of the temperature sensors used for taking the measurements in our experiments.

The results presented in this article summarize our initial experience in the area of application-specific thermal energy modeling of current processors. Several open research issues and topics are subject of current and future work. Some of them include a thermal modeling methodology applicable to real application codes and scalability studies at node, rack, and data center level. We are developing further our initial results towards algorithms and mechanisms for analyzing and improving the application-specific balance between temperature, power, and performance. For example, we are planning to use our thermal model for building smart scheduling and dynamic runtime control algorithms in order to identify energy efficient operating point depending on the properties of the application code.

6. Acknowledgements

We gratefully acknowledge the support and resources allocated at PNNL for the installation and operation of the SeaPearl cluster – a unique instrumentation and monitoring facility which enabled the experiments presented in this article.

References

[1] V. Getov, A. Hoisie and P. Bose, New Frontiers in Energy-Efficient Computing, *IEEE Computer* **49**(10), 14-18, IEEE Press, 2016.

[2] Y.A. Cengel and A.J. Ghajar, *Heat and Mass Transfer: Fundamentals and Applications*, 5th Edition, McGraw Hill, 2015.

[3] V. Getov, D.J. Kerbyson, M. Macduff and A. Hoisie, Towards an Application-Specific Thermal Energy Model of Current Processors, *Proc. E2SC2015*, 1-10, ACM Press, 2015.

[4] F.W. Kussy and J.L. Warren, *Design Fundamentals for Low-Voltage Distribution and Control*, Chapter 2 Heat Transfer in Electrical Components, 31-62, CRC Press, 1986.

[5] E. Rotem, R. Ginosar, A. Mendelson and U.C. Weiser, Power and Thermal Constraints of Modern System-on-a-chip Computer, *Proc. 19th Int. Workshop on Thermal Investigations of ICs and Systems (THERMINIC)*, 141-146, IEEE Press, 2013.

[6] WattsUp? *Power Analyzer, Watt Meter and Electricity Monitor, Operators Manual*, Revision 9, 2008. https://www.wattsupmeters.com/secure/downloads/manual_rev_9_corded0812.pdf

[7] Intel®, *64 and IA-32 Architectures Software Developer's Manual, Volume 3: System Programming Guide*, 2016. http://www.intel.com/content/dam/www/public/us/en/documents/manuals/64-ia-32-architectures-software-developer-system-programming-manual-325384.pdf

[8] J.H. Laros III, P. Pokorny and D. Debonis, PowerInsight – A Commodity Power Measurement Capability. *Proc. Int. Green Computing Conference, (IGCC)*, 1-6, IEEE Press, 2013.

[9] P. Hammarlund, Haswell: The Fourth-Generation Intel Core Processor, *IEEE Micro* **34**(2), 6-20, IEEE Press, 2014.

[10] Intel®, *Technical Product Specification for Server Board S2600CO Family*, Revision 1.7, 2015. http://www.intel.com/content/dam/support/us/en/documents/motherboards/server/s2600co/sb/g42278004_s2600co_tps_rev171.pdf

[11] D. Hackenberg, R. Oldenburg, D. Molka and R. Schone, Introducing FIRESTARTER: A Processor Stress Test Utility, *Proc. Int. Green Computing Conference, (IGCC)*, 1-9, IEEE Press, 2013.

[12] Argonne National Laboratory, *NEKBONE: Thermal Hydraulics Mini-Application*, Release 3.1, 2013. https://cesar.mcs.anl.gov/sites/cesar.mcs.anl.gov/files/nekbone_3.1_readme.pdf

[13] ASHRAE Technical Committee 9.9, *Thermal Guidelines for Data Processing Environments*, 3rd Edition, ASHRAE Datacom Series, ASHRAE, 2012.

[14] D.J. Kerbyson, A. Vishnu and K.J. Barker, Energy Templates: Exploiting Application Information to Save Energy, *Proc. IEEE CLUSTER 2011*, 225-233, IEEE Press, 2011.

[15] J.H. Laros III, K.T. Pedretti, S.M. Kelly, W. Shu, K.B. Ferreira, J. Van Dyke and C.T. Vaughan, *Energy-Efficient High Performance Computing – Measurement and Tuning*, Springer Briefs in Computer Science, Springer, 2013.