Granularity of Microprocessor Thermal Management: A Technical Report

[†]Karthik Sankaranarayanan, [†]Wei Huang,
[‡]Mircea R. Stan, [§]Hossein Haj-Hariri, [§]Robert J. Ribando and [†]Kevin Skadron
†*Department of Computer Science,*
‡*Charles L. Brown Department of Electrical and Computer Engineering,*
§*Department of Mechanical and Aerospace Engineering,*
University of Virginia, Charlottesville, VA
{ks4kk,skadron}@cs.virginia.edu, {mircea,hh2b,rjr}@virginia.edu

Abstract

Process technology scaling, poor supply voltage scaling and the resultant exponential increase in power density have made temperature a first-class design constraint in today's microprocessors. An interesting question in the context of thermal management and multi-core architectures is about the correct size granularity of thermal management. It is known that the silicon substrate acts as a spatial low-pass filter for temperature. This means that if blocks with very high power density are small enough (for e.g.*, if they are below a "cut-off" size), they do not cause hot spots. This paper investigates this phenomenon analytically and presents a discussion through three microarchitectural examples. First is a thermal study of a many-core architecture which illustrates the thermal benefit of many small cores as opposed to a few large cores. This study also explores the effect of local* vs. *global thermal management. Second is an investigation of whether high aspect ratio sub-blocks such as cache lines can become hot spots due to pathological code behaviour. Third is an exploration of thermal sensor accuracy as a function of the number of sensors and a characterization of two sensor interpolation schemes as a means to reduce sensor errors.*

1 Introduction

Two important recent trends have had a tremendous impact on the microprocessor industry: First, non-ideal semiconductor technology scaling has made physical effects that were previously abstracted away by computer architects to become first-order design constraints. A crucial example of this is the exponential increase of power density as feature size shrinks. This has caused temperature to be a serious concern since power density is directly related to on-chip temperature. Second, multi-core processors have become the norm because of the disproportionate scaling of wire and logic delays, diminishing Instruction-Level Parallelism (ILP) and the challenges of power and heat dissipation. In such a multi-core era, an interesting question is the size granularity at which microarchitectural thermal management should be performed. That is, if thermal management is most relevant when performed at the level of a sub-block (*e.g.* cache line, register file entry *etc.*), functional block (*e.g.*, register file, cache, branch predictor *etc.*), architectural sub-domain within a core (*e.g.*, the integer pipeline, the floating point pipeline, the memory pipeline *etc.*), at the level of a core, or globally for the entire chip.

1.1 Granularity of Thermal Management

Temperature-aware design can be approached in two steps. The first is to stop the wasteful power consumption (which then gets converted into heat dissipation) at the source through best-effort power-aware design. However, caution has to be exercised in employing power-saving schemes that can make *power density* worse, thereby increasing the temperature. Once the heat dissipation has been minimized as much as possible through effective power management, the next step is to distribute the given amount of heat well to prevent it from being converted into peak temperature. In the strictest sense, only this latter step can be called microarchitectural *temperature* management since the former is actually temperature-aware *power* management. In such terms, global Dynamic Voltage and Frequency Scaling (DVFS), which scales the supply voltage and the clock frequency of the entire chip together, is mainly a power management scheme as it performs only the first step above. Nevertheless, it is hard to beat as a Dynamic Thermal Management (DTM) scheme because of many reasons: a) The voltage + frequency knob provides cubic savings in power for a linear reduction in delay; b) A factor of k reduction in frequency results in $< k$ reduction in performance (*e.g.* memory bound applications spend fewer CPU cycles waiting); c) Thermal time constants are larger than the typical microarchitectural timescale of kilo-instructions and block-level architectural DTM quickly degenerates into a global scheme and d) As the size of a core reduces due to scaling, a sub-domain within a core is affected more by its neighbour's temperature than its own computational activity. So, neighbouring sub-domains effectively get aggregated for effective thermal management, again resulting in a global scheme.

Investigation of this last phenomenon shows that the determining factor in it is the relationship between heat conduction in the vertical (through the die, heat spreader and heat sink) and lateral (along the die) directions. When the former is much greater than the latter, a block's temperature is mainly dependent on its own computational activity. Otherwise, it is mainly neighbour influence. Technology scaling results in lateral dimensions of silicon diminishing faster than the vertical thickness for reasons of mechanical strength. Moreover, vertical dimensions of the package layers (heat spreader, heat sink) do not scale much. This means lateral spreading grows quicker than vertical conduction. The implication is that the spatial granularity of thermal management has to become coarser with technology scaling. One would have to move from thermal management at the level of a functional block to the core-level to groups-of-cores. With the advent of multi-cores and many-cores, how this happens is an interesting question for exploration.

1.2 Spatial Filtering

It turns out that silicon behaves like a spatial low-pass filter for temperature. That is, power dissipating blocks that are small enough do not cause hot spots. Silicon "filters out" these "spatial high frequencies" (blocks of small size). Figure 1 illustrates this aspect. The image on the left (1(a)) shows a silicon die with two square power dissipating blocks. Both the blocks have the same power density ($1\frac{W}{mm^2}$) but the area of the block on the left is four times that of the one on the right. Although the power densities are the same, it can be seen that the larger block on the left is significantly hotter than the smaller block on the right. This relationship between block-size and peak temperature is illustrated in figure 1(b). It is the classic "Bode plot" frequency response of a low-pass filter with the spatial frequency (1/block-size) plotted on the x-axis and the peak temperature on the y-axis. Both the axes are on a logarithmic scale. It shows a clear "cut-off frequency" beyond which the temperature falls rapidly.

A study of this spatial filtering effect from a microarchitectural perspective is useful since it throws light on several aspects of the size granularity of thermal management. Hence, this paper explores this thermal spatial filtering effect in detail. Since the effect is nothing but a manifestation of the relationship

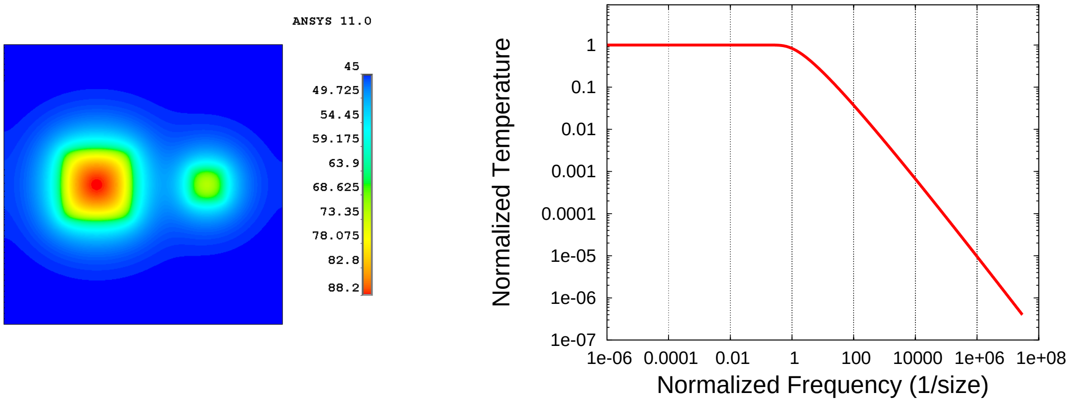


Figure 1. Illustration of the spatial thermal filtering. (a) Two blocks of different sizes having the same power density. (b) low-pass filter "Bode plot" frequency response.

between lateral and vertical heat conduction in silicon, it first takes an analytical approach and solves the two-dimensional steady state heat equation from first principles for a sample geometry similar to that of microprocessor chips but in two dimensions. Then, in the light of the analytical equation, it explores how the spatial filtering appears in practice when the assumptions of the analytical model are removed. Finally, it illustrates the spatial filtering behaviour using three microarchitectural examples:

1. A many-core checkerboarding experiment where the die is tiled in a checkerboard-like fashion with alternating cores and L2 blocks. The objective is to study the temperature by varying the number of cores keeping the total area occupied by the cores, the power dissipated in them and their power density constant. It illustrates the potential increase in Thermal Design Power (TDP) achievable solely due to spatial filtering when all the other variables remain the same. Furthermore, it studies the relationship between the number of cores and the granularity of thermal management by exploring the effectiveness of local (within-core) thermal management as opposed to global (core-level) thermal management as a function of the number of cores.

2. The question of whether a high aspect ratio sub-block like a single cache line or a register file entry can become the hot spot because of malicious code behaviour is examined from a spatial filtering perspective. This illuminates the relationship between aspect ratio of a block and its peak temperature.

3. For a regular grid of thermal sensors, this paper studies the relationship between the inter-sensor distance and the error in temperature measurement and explains it using spatial filtering. It also explores the effectiveness of interpolation schemes in mitigating sensor errors.

The remainder of this paper is organized as follows. Section 2 discusses the previous work related to this paper. Section 3 describes the analytical formulation and solution of the two-dimensional heat equation for boundary conditions similar to those in microprocessors. It also reconciles the analytical model with what is observed in practice. Section 4 details the microarchitectural examples of the spatial filtering effect and presents the results. Section 5 concludes the paper providing pointers to interesting future directions.

2 Related Work

Three papers we know have examined the spatial filtering aspect from the viewpoint of thermal modeling accuracy. Etessam-Yazdani *et. al.* [7] study the temperature response of the chip to a white noise power distribution experimentally. They also perform a two-dimensional Fast Fourier Transform (FFT) analysis to determine the spatial cut-off frequency beyond which high power density has lesser impact on temperature. They conclude that for accurate thermal modeling, it is sufficient to model at a size granularity close to this cut-off frequency. Two previous papers from our group [10, 8] also deal with the correct spatial granularity for modeling temperature accurately. In studying the size granularity, [10] employs an equivalent electrical approximation through a resistor network analogy. [8] examines the impact of high aspect ratio functional blocks on thermal modeling accuracy and sub-divides high aspect ratio blocks into multiple sub-blocks with aspect ratios closer to unity. These papers mainly approach the size granularity from the standpoint of thermal *modeling* accuracy and not from a thermal *management* perspective. They do not study the microarchitectural implications of thermal filtering. Furthermore, they do not include the effect of the package in detail, which, this paper shows to be an important factor.

Another recent paper from our group that deals with the spatial filtering aspect is [9]. It discusses the thermal efficiency of many-core processors in the context of spatial thermal filtering. While it has a microarchitectural angle with the consideration of many-core design, its focus is narrow as it only approaches a single design aspect related to thermal filtering — core size and TDP benefit. In comparison, our current paper expands on it and considers broader implications of spatial filtering on microarchitectural thermal management including aspects such as granularity of thermal *control* and sensor accuracy. Moreover, even for many-core processors, the results presented in the current paper are more detailed, taking into account the effect of the package, chip thickness, area of the L2 cache *etc.* Also, like [10], [9] reasons about spatial filtering using an equivalent RC approach. This paper, on the other hand, studies the thermal conduction equation directly. Furthermore, none of the above previous papers except [8] consider the effect of aspect ratio, which is nothing but an alternative manifestation of the spatial filtering.

Since this paper treats the accuracy of thermal sensors in the light of spatial filtering, research on sensor accuracy and fusion is relevant here. Lee *et. al.* [13] present an analytical model for determining the sensor accuracy as a function of the distance from the hot spot. However, they do not offer any techniques for improving sensor accuracy. Memik *et. al.*'s work on sensor allocation [14] discusses uniform and non-uniform placement of sensors. For uniform sensor allocation, it describes a simple interpolation scheme to reduce sensor errors. The interpolation schemes proposed in this paper are more accurate than the simple scheme from [14] but without significantly higher overhead. Sharifi *et. al.* [17] employ Kalman filter for eliminating sensor errors. While their scheme is very accurate, it requires the knowledge of per-unit power consumption numbers, which might not be available easily. The usefulness of our work is in offering an option *between* a simple scheme such as [14] and a computationally intensive scheme such as Kalman filtering [17].

Dadvar and Skadron [5] raise concerns about the possible security risks involved in software-controlled thermal management. Since our work examines the question of pathological code heating up individual cache lines, [5] is relevant here. Our finding is partly in agreement with [5] in that there is a possibility of cache lines heating up to high temperatures due to code activity. However, their aspect ratios prevent the temperatures from reaching catastrophic limits. Ku *et. al.*'s research [12] on reducing the power density of caches by keeping the live lines as far apart from each other as possible is relevant to this work as well. In effect, they exploit the spatial filtering by increasing the spatial frequency. Our work offers a framework to reason about such thermal management schemes.

3 An Analytical Approach

As explained in section 1, at the core of the thermal filtering is the relationship between heat conduction in the lateral direction along the die and the vertical direction through the die, heat spreader and heat sink. In order to understand this relationship better, an analytical look into a simplified version of the problem is beneficial. While the heat conduction, geometry and boundary conditions of an actual chip and package are quite complex, we make several simplifying assumptions in this section for the sake of mathematical ease. However, we do examine the effect of these assumptions later by comparing the trend predicted by the analytical model to experimental data obtained by simulating a realistic chip and package configuration in a Finite Element Model (FEM).

3.1 A Simplified Heat Conduction Problem

The assumptions made in this section are the following:

- **Simplification of the geometry:** In an actual chip, heat is conducted in all three dimensions. However, since the two lateral dimensions are not conceptually different from each other, collapsing them into one will not affect the nature of the solution qualitatively. Hence, we consider a two-dimensional version of the heat conduction problem with one vertical dimension and one lateral dimension as opposed to the full three-dimensional case. In fact, we actually examined a preliminary solution of the full three dimensional case and observed that its analytical form is not qualitatively different from that of the 2-D case. Hence, in the interest of time and clarity, we present the 2-D solution here. Moreover, for mathematical tractability, we only consider a single vertical layer of conduction (silicon) and ignore the other package and interface layers. Although this is the most serious assumption, making it affords a much simpler treatment of the problem. Also, it is not far from reality for mobile processors where the form factor and cost preclude any cooling solution.

- **Simplification of the boundary conditions:** We specifically make three assumptions. First, we assume an isothermal boundary at the top (the end exposed to the ambient temperature). In real processors, this is the top of the heat sink fins where the fan resides. Hence, this boundary is convective in reality. Second, we assume that the bottom of the chip (the power dissipating end) is insulated and hence no heat flows through the I/O pads and the Printed Circuit Board (PCB). Third, we assume the boundaries of the lateral dimension to extend indefinitely. This is also done for mathematical convenience as infinite boundaries provide simple closed-form solutions comprising of exponentials.

Furthermore, we perform only a steady state analysis here. Transient analysis would include an extra temporal dimension which would make it more complex than necessary. For peak power density, steady state temperature provides an upper bound on the attainable temperature and hence is sufficient for our purpose.

Figure 2 represents the above-mentioned simplifications pictorially. Figure 2(a) can be thought of as representing a sheet of silicon with a thickness l (along the y-axis) that extends indefinitely along the positive and negative x-axes. Its depth (along the z-axis) is considered negligible, effectively reducing the problem to two dimensions. A power-dissipating block of size $2a$ with a power density of q resides at the bottom surface of silicon. The top surface of silicon is isothermal with a temperature of 0 (when the ambient temperature is non-zero, it only shifts the solution by that constant value). Our task is to find the temperature distribution on this infinite sheet of silicon as a function of a and l. This is a classic heat transfer problem that can be solved by using standard Partial Differential Equation (PDE) solving techniques [4, 15, 11].

Assuming that the origin is at the center of the power dissipator, it can be seen that the setup shown in figure 2(a) is symmetric about the y-axis. Therefore, the left and right halves are indistinguishable with respect to temperature. Hence, the net heat flux (power density) from one side of the y-axis to the other must be zero. If not, that very difference constitutes an asymmetry through which the two halves can be distinguished. Thus, the y-axis behaves like it is insulated. This is shown in figure 2(b), which is essentially the right half of 2(a) with slanted lines clearly marking the insulated behaviour of the y-axis.

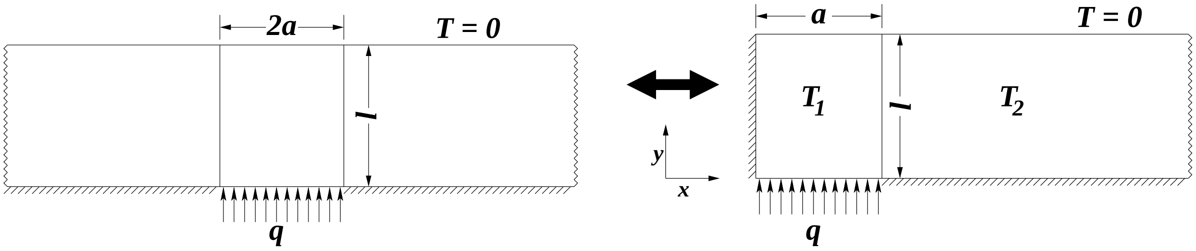


Figure 2. Simplified heat conduction problem to mimic a silicon die of thickness l. A power density of q is applied at the bottom in a region of size $2a$. Insulated boundaries are marked by slanted lines. Serrated lines indicate that the boundaries extend to infinity in that direction. (a) shows the original infinite version of the problem and (b) shows the equivalent semi-infinite version that (a) reduces to because of symmetry. (b) is essentially the right half of (a)

.

3.2 Analytical Solution

For a moment, let us consider this heat conduction problem qualitatively. In steady state, the temperature distribution of this infinite block does not change over time. This means that for any infinitesimal volume in it, there is no change in its internal energy. In other words, the rate at which heat energy enters it (input power) from its neighbours through conduction is the same as that of the heat energy leaving it (output power). Another way of saying this is that the *gradient* (derivative in space) of the power (and hence the power density, since area is constant) is zero. The relationship between this power entering and leaving the volume and its temperature is governed by Fourier's law of heat conduction [4] which is nothing but the "Ohm's law" for thermal conduction. It says that the power density is directly proportional to the gradient of the temperature and that the constant of proportionality is the thermal conductivity of the material (in our case, silicon). This is intuitive because higher the gradient, higher is the difference between the temperatures of this volume and its neighbours and hence, higher is the rate of heat transfer. So, the fact that in steady state, the gradient of the power density is zero can be re-stated using Fourier's law that the *gradient of the gradient of temperature* is zero. This statement with a *second order* derivative in it is called the steady state heat equation, which we will be solving for the two-dimensional case.

If we denote the steady state temperature functions of the two portions of figure 2(b) by $T_1(x, y)$ and $T_2(x, y)$ and the gradient operator by ∇ (for two dimensions, this is nothing but $\frac{\partial}{\partial x} + \frac{\partial}{\partial y}$), the steady state heat equations we would like to solve are:

$$\begin{array}{ccc} \nabla^2 T_1 = 0 & , & \nabla^2 T_2 = 0 \\ (0 \leq x \leq a, 0 \leq y \leq l) & & (a \leq x \leq \infty, 0 \leq y \leq l) \end{array} \qquad (1)$$

Now let us list the boundary conditions for our problem. From the discussion above, if k is the thermal conductivity of silicon, Fourier's law is nothing but the fact that *power density* $= -k\nabla T$. Hence, for an insulated boundary, since the power density across the boundary is zero, the condition can be written as $\nabla T = 0$. Thus, the boundary conditions for our problem are:

$$
\begin{aligned}
T_1(x,l) &= 0 \qquad T_2(x,l) = 0 && && \text{(2a)}\\
& \qquad\quad T_2(\infty,y) = 0 && (\textit{isothermal boundaries}) && \text{(2b)}\\
\frac{\partial T_1}{\partial x}|_{x=0} &= 0 \qquad \frac{\partial T_2}{\partial y}|_{y=0} = 0 && (\textit{symmetry, insulation}) && \text{(2c)}\\
\frac{\partial T_1}{\partial y}|_{y=0} &= -\frac{q}{k} && (\textit{Fourier's law, power dissipator}) && \text{(2d)}\\
T_1(a,y) &= T_2(a,y) && && \text{(2e)}\\
\frac{\partial T_1}{\partial x}|_{x=a} &= \frac{\partial T_2}{\partial x}|_{x=a} && (\textit{continuity at } x=a) && \text{(2f)}
\end{aligned}
$$

The analytical solution of this boundary value problem can be obtained using classic PDE-solving techniques [4, 15, 11]. We provide only the closed-form solutions of the functions T_1 and T_2 here and refer interested readers to appendix 5 for a detailed derivation.

$$
T_1(x,y) = \frac{ql}{k}\left[1 - \frac{y}{l} - 2\sum_{n=0}^{\infty}\frac{e^{-(\gamma_n \frac{a}{l})}}{\gamma_n^2}\cosh(\gamma_n\frac{x}{l})\cos(\gamma_n\frac{y}{l})\right] \tag{3}
$$

$$
T_2(x,y) = \frac{ql}{k}\left[2\sum_{n=0}^{\infty}\frac{\sinh(\gamma_n \frac{a}{l})}{\gamma_n^2}e^{-(\gamma_n \frac{x}{l})}\cos(\gamma_n\frac{y}{l})\right] \tag{4}
$$

$$
\textit{where,} \quad \gamma_n = (2n+1)\frac{\pi}{2} \quad (n = 0, 1, 2, \ldots)
$$

It is useful to make a couple of observations about these solutions. It can be seen that all the variables of interest (x, y and the half-size a) appear in the equation as ratios with respect to l. This indicates that the lateral dimensions matter only relative to the vertical thickness and not in the absolute. For our purposes, it is sufficient to restrict ourselves to the peak temperature on the infinite sheet. Actually, for a given a, the peak temperature T_{peak} occurs at $(0,0)$ *i.e.*, at the center of the power dissipator. Hence, $T_{peak} = T_1(0,0)$. Also, the absolute maximum peak temperature (for all a) occurs when $a = \infty$ *i.e.*, when the power dissipating block is very large. Let us call this temperature T_{peak}^{∞}. Then, $T_{peak}^{\infty} = \frac{ql}{k}$, where q is the power density and $\frac{l}{k}$ divided by the area of the power dissipator gives the vertical *thermal resistance*. Now, defining the normalized peak temperature T_{peak}^{norm} as $T_{peak}/T_{peak}^{\infty}$ and the normalized half-size a_{norm} as $\frac{a}{l}$, we get

$$
\begin{aligned}
T_{peak}^{norm} &= \frac{T_{peak}}{T_{peak}^{\infty}} = \frac{T_1(0,0)}{\frac{ql}{k}}\\
&= 1 - \frac{8}{\pi^2}\sum_{n=1,3,5,\ldots}\frac{e^{-n\frac{\pi}{2}a_{norm}}}{n^2}
\end{aligned} \tag{5}
$$

It is this T_{peak}^{norm}, whose value ranges between 0 and 1, that has been plotted against $\frac{1}{a_{norm}}$ in figure 1(b). One can clearly see the low-pass filter behaviour with a cut-off frequency around $a_{norm} = 1$. This means that when the half-size of the power dissipator is smaller than the vertical thickness, the peak temperature falls rapidly in relation to the half-size.

3.3 Spatial Filtering in Practice

Equation (5) and figure 1(b) show the exponential relationship between the size of a power dissipator and its peak temperature. Similarly, (3) shows the linear relationship between the power density q and peak temperature. This suggests an interesting trade-off for microarchitectural thermal management. A thermal management scheme can reduce the temperature of a hot block by reducing its computational activity, thereby reducing its power density. Alternatively, it can manipulate the size of the power dissipating block (*e.g.*, by partitioning the block and separating the partitions by a safe distance) and hence exploit this spatial filtering effect to reduce peak temperature. When the size of the power dissipating block is close to the cut-off frequency, this latter choice can have a significant benefit over the former because of its exponential relationship as opposed to the linear relationship in the former. At the same time, when the block is so large that even the partitions are much larger than the cut-off frequency, reducing the power density might be a better choice since the benefit from size reduction becomes negligible. In order to examine such microarchitectural implications, it is necessary to reconcile the theoretical model discussed in the previous section with what happens in practice. So, in this section, we relax the assumptions of the analytical model and examine the effects through experiments under a commercially available FEM solver, ANSYS 11.0 [1].

3.3.1 Impact of Size

We remove the assumptions of the last section in three steps. First, we consider the full three-dimensional problem with a convective boundary near the ambient temperature. Next, we include the effect of multiple package layers and finally, we investigate the impact of geometric boundaries. For the first step, we consider a 10 mm x 10 mm silicon die that is 1 mm thick. The top of the die is cooled by convection to an ambient at $0°C$ with a heat transfer co-efficient equivalent to a 0.1 $\frac{K}{W}$ thermal resistance (*i.e.*, for a $100mm^2$ area, it is $0.1\frac{W}{mm^2K}$) . At the center of the base of the die is a square power dissipator with a power density of $1\frac{W}{mm^2}$. The size of this power dissipator is varied from 0.2 mm to the full 10 mm. The maximum peak temperature for this experiment occurs when the power dissipator occupies the entire silicon die and its value is 20 degrees more than the ambient. We call this experiment *center-Si* to denote the single layer of *silicon* with a power dissipator at its *center*.

Next, we repeat the same experiment with the die attached to three other layers — a layer of Thermal Interface Material (TIM) followed by a layer of copper (representing the heat spreader), followed by a second layer of copper (denoting the heat sink). To isolate the effect of multiple layers from that of finite boundaries, we restrict the lateral dimensions of all the four layers to 10 mm x 10 mm. The vertical thickness of each layer is set in such a manner that the *proportion* of the thicknesses is similar to what is seen in a typical package and that the total *equivalent thickness* in silicon terms (*i.e.*, sum of the thicknesses of the individual layers weighted by the ratios of their thermal conductivities to that of silicon) is still 1 mm. So, the maximum peak temperature is still $20°C$. The actual thicknesses of the silicon, TIM, spreader and heat sink layers are 0.0571 mm, 0.0076 mm, 0.3810 mm and 2.6286 mm respectively. The respective thermal conductivities are 100, 4, 400 and 400 $\frac{W}{mK}$. The heat transfer co-efficient at the top surface remains the same as before and the size of the power dissipator is varied as before. We call this experiment *center-4layers-equal* to denote the equal lateral size of the four layers and the location of the power dissipator at the center of the base.

Finally, to examine the effect of the boundaries, we add a few more configurations. To model a normal package in which the heat spreader and heat sink are larger than the die, we extend the *center-4layers-equal* configuration to make the lateral sizes of the heat spreader and the heat sink layers to be 20 mm x 20 mm and 40 mm x 40 mm respectively. We call this *center-4layers-spread* since it models the effect of lateral spreading beyond the die area into the spreader and the sink. The heat transfer co-efficient remains the same as before. It is to be noted that the maximum peak temperature in this case will be lower than the $20°C$ seen in the previous cases because of the lateral spreading beyond the die area. Finally, to study the effect of the location of the power dissipator (and hence the effect of finite boundaries), we extend all of the above configurations by changing the location of the power dissipator from the center of the base to a corner. This restricts the directions in which heat spreads laterally in silicon to two from the four possibilities (north, south, east and west) at the center. We name these configurations in the form *corner-xxxx* where *xxxx* derives from the names of the previous configurations. For instance, *corner-Si* is basically the same as *center-Si* except for the location of the power dissipator and so on. A *center-xxxx* experiment and the corresponding *corner-xxxx* experiment can be thought of as forming the lower and upper bounds on the attainable peak temperature in that particular configuration for a given power density.

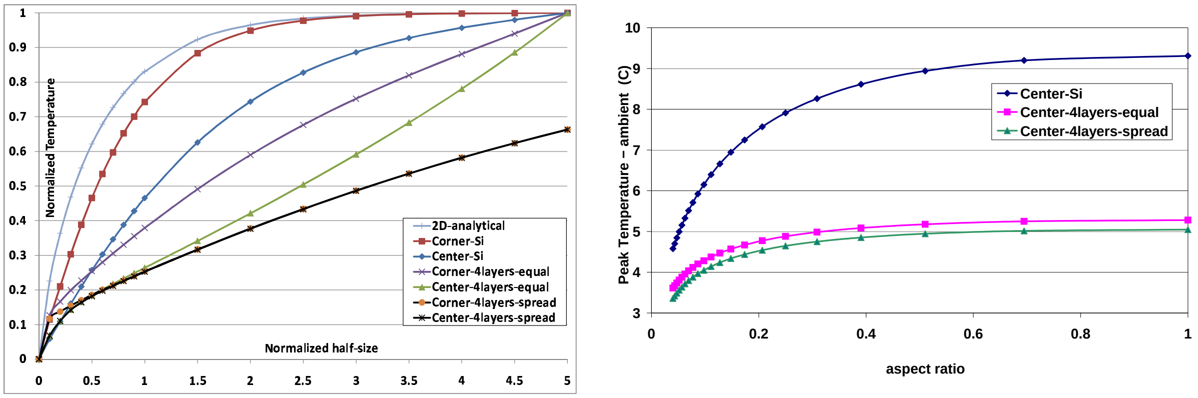


Figure 3. Comparison of the analytical model and FEM simulation. (a) shows the impact of power dissipator size and (b) shows the effect of aspect ratio.

All the above experiments are run under ANSYS 11.0 with a lateral grid size of 100 x 100 on silicon (except *corner-4layers-spread*, which uses a 50 x 50 grid due to a software license restriction on the number of nodes). The lateral dimensions of the grid cells are the same in the other layers as well. Vertically, the silicon die is divided into three layers, spreader and sink are divided into four layers each and the TIM is modeled as a single layer. Figure 3(a) shows the results of these experiments. It plots the peak temperature of a configuration as a function of the size of its power dissipator. Both the axes are normalized. The temperature axis reports the ratio of the peak temperature of a configuration to the maximum peak temperature of *center-Si*. The size axis reports the ratio of the half-size of the power dissipator to the vertical thickness of the configuration (1 mm in silicon equivalent). For comparison, it also plots the peak temperature computed by the analytical equation (5) (*2d-analytical* in the graph).

The main conclusion one can draw from the graph is that the behaviour suggested by the analytical equation is not universal. There is a significant difference between the exponential relationship suggested by the analytical equation and the "straight line" plots of the *4layers-spread* configurations that model a desktop-type package. The configurations with only a single layer of silicon (*corner-Si* and *center-Si*) on the other hand, are closer to the analytical equation showing that going from two to three dimensions or from an isothermal boundary to a convective boundary are not factors causing significant difference.

Furthermore, it can be seen that the plot shows clear “bands” bounded from below by a *center-xxx* configuration and from above by the corresponding *corner-xxx* configuration. This means that the location of the power dissipator matters to the degree indicated by the width of the “band”. Hence, the finite boundaries of the geometry affect different configurations differently. The location matters a lot for a mobile-type configuration without any spreader or sink (**-Si*) while it does not matter at all for a typical desktop-type package (**-4layers-spread*). This is because of the extra heat spreading offered by copper (as opposed to silicon) and the extra area available for spreading beyond the die. Since the entire die is roughly at the center of the heat spreader and the heat sink, the position of the power dissipator within the die becomes immaterial.

Thus, the lateral heat spreading in the heat spreader and the heat sink are the most significant factors determining the extent of spatial thermal filtering. The steep exponential nature of the spatial filtering indicated by the analytical equation is only valid for chips without a cooling solution (*e.g.* processors in hand-held devices). For a typical package found in desktop processors, the exponential is so shallow that it behaves like a straight line. Since previous work [7, 10] examined the spatial filtering from the standpoint of thermal *modeling* accuracy, it was sufficient for them to consider the “worst-case” thermal gradients that occur in the absence of a heat spreader or a heat sink. However, for a thermal *management* angle, the distinction is indispensable.

3.3.2 Impact of Aspect Ratio

In the analytical simplification of the spatial filtering, a factor that was not considered due to the lack of three dimensions was the effect of the lateral aspect ratio of the power dissipator on its peak temperature. This relationship is important to understand the thermal behaviour of high aspect ratio microarchitectural sub-blocks like cache lines. Hence, figure 3(b) plots this relationship for the three *center-** configurations. The area of the power dissipator is fixed at $4mm^2$ and its aspect ratio is varied from 1:25 to 1:1. At an aspect ratio of 1:1 (2 mm x 2 mm block ⇒ half-size = 1 mm), this is nothing but the set of points corresponding to the normalized half-size of 1 in figure 3(a). It can be seen from the plot that aspect ratio also has an exponential relationship with peak temperature. Also, similar to the dissipator size, the extent of spatial filtering is dependent on the additional layers of copper in the package. The extra heat spreading in copper essentially smoothes out the response of the thermal filter. Hence, the curves are shallow for the *4layers* configurations and steep for *Center-Si*.

4 Microarchitectural Examples

So far, we have examined the thermal spatial filtering phenomenon in isolation. In this section, we will study its microarchitectural implications through three microarchitectural examples. The first is a study of the thermal efficiency of many-core processors in the light of the granularity of thermal management. The second is an investigation of whether high aspect ratio sub-blocks like cache lines can become hot spots due to pathological code behaviour. The third is an exploration of thermal sensor accuracy as a function of the distance between sensors and an examination of the effectiveness of sensor interpolation schemes.

4.1 Many-Core Processors

4.1.1 Thermal Benefit of Spatial Filtering

As it was mentioned in section 1, thermal management, when considered distinctly from power management, is mainly the efficient distribution of heat within the available die area. One way to accomplish this is through a cooling system that spreads heat well. For instance, heat spreaders made out of artificial diamonds are available commercially [20], since diamond has the highest thermal conductivity among known materials. Heat pipes are also an example of such efficient spreading. On the other hand, considering the thermal spatial filtering discussed before, many-cores provide an interesting alternative. In comparison with large monolithic cores, they provide an opportunity to distribute the *heat generation* better. Since the power density on the lower level caches is much lower compared to that on the cores, a possibility is to floorplan the die in the form of a checkerboard with cores forming the dark squares and lower level cache banks forming the light squares (see figure 4(a) for an example). This exploits the spatial filtering by maximizing the spatial frequency of the power density distribution. Hence, in this section, we study the effect of such checkerboarding of many-cores in the light of spatial filtering.

Assuming that the silicon die is arranged in the form of a checkerboard as shown in figure 4(a), we vary the number of cores by varying the size of the checkerboard (from 2 x 2 to 20 x 20) and study its effect on peak temperature. The power densities in the cores and the cache blocks are assumed to be constant nowadays through the scaling of the number of cores. This is done to isolate the thermal effect of spatial filtering from other circuit and microarchitectural factors affecting temperature. The constant power density assumption is also reasonable under constant electric field scaling principles [6], since the microarchitecture of a core is assumed to remain constant while its size is scaled as permitted by technology. Hence, with the number of cores increasing, the total *power* dissipated remains constant across the configurations. Since the thickness of the die does not scale with the feature size for reasons of mechanical strength, it is assumed to be constant as well. Later, we examine the effect of this assumption by performing a sensitivity study on varying the die thickness.

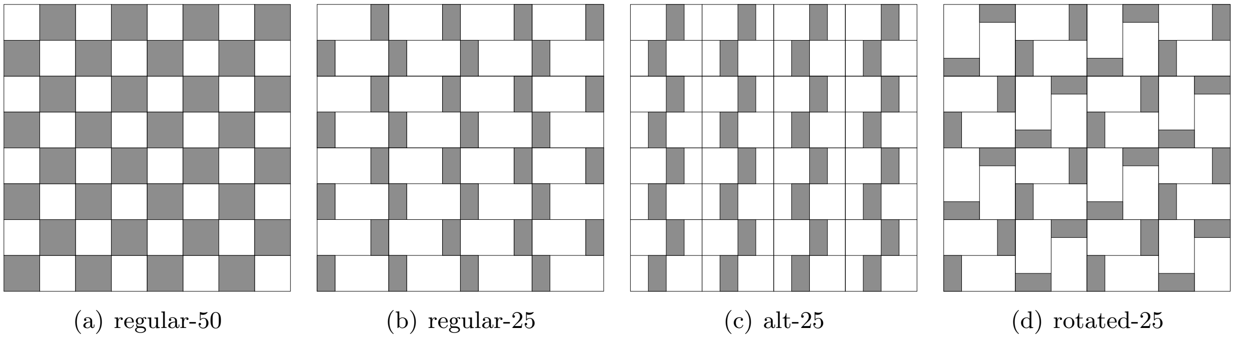
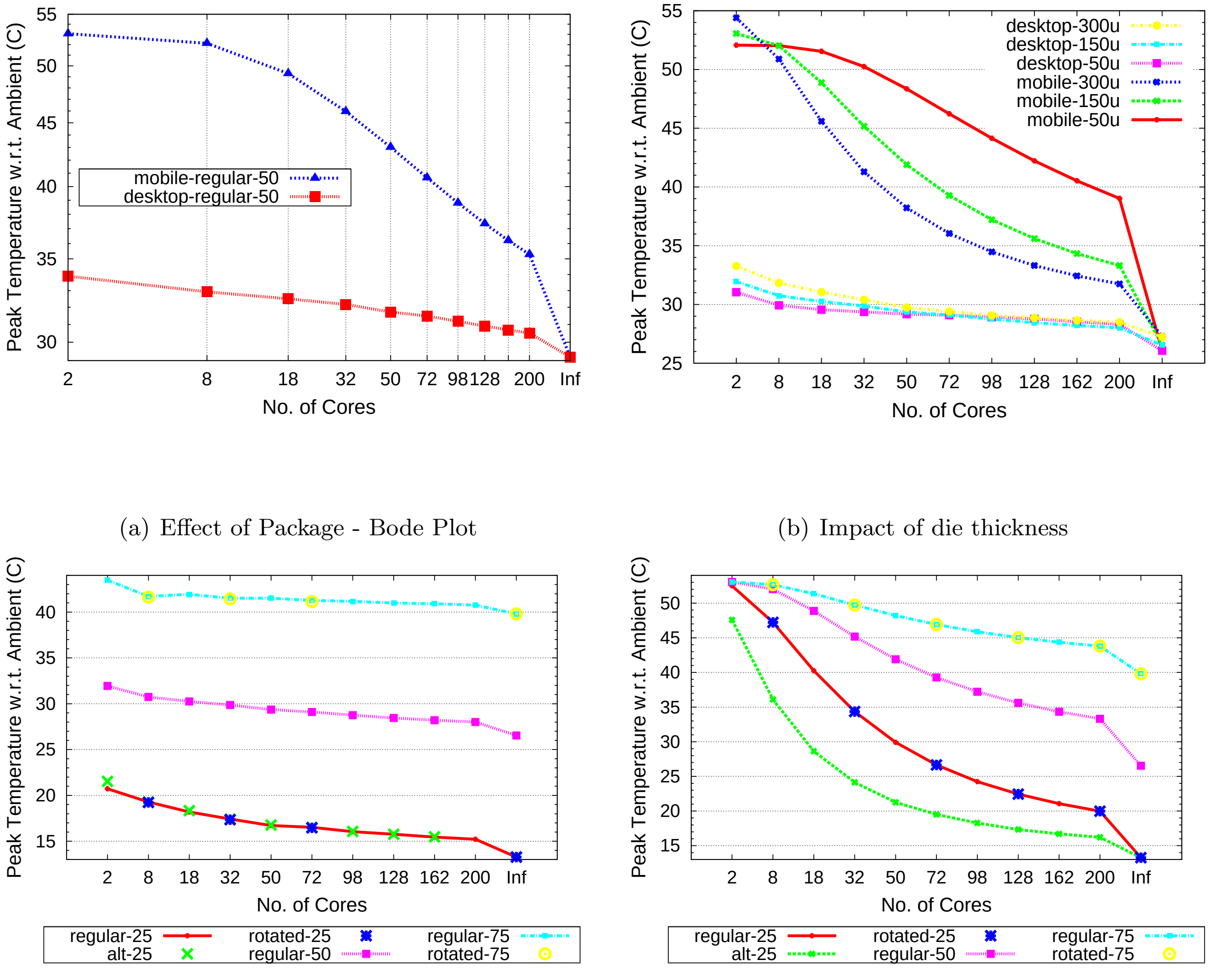

(a) regular-50 (b) regular-25 (c) alt-25 (d) rotated-25

Figure 4. Illustration of the several checkerboard configurations studied. Darkly shaded areas denote the cores and the unshaded areas denote the lower level cache banks. The numeric suffix attached to the name of each configuration indicates the percentage of the die area occupied by the cores. Each figure above shows a many-core die with 32 cores.

We know from the discussion in the last section that the package is a significant determinant in the spatial filtering. Hence, we consider two different package configurations: one without any cooling system to mimic hand-held mobile processors and another with a typical package found in desktop processors with a heat spreader and a heat sink. We call the former *mobile-regular-50* with *regular-50* denoting a regular checkerboard as shown in figure 4(a), with the cores occupying 50% of the total die area. In a similar vein, the latter is called *desktop-regular-50*.

The experimental setup is as follows: the silicon die is 16 mm x 16 mm x 0.15 mm in size for both cases. For *desktop-regular-50*, the additional layers include a TIM of thickness $20\mu m$ and the same lateral dimensions as the die, a copper heat spreader of size 3 cm x 3 cm x 1 mm and a copper heat sink of size 6 cm x 6 cm x 6.9 mm. The heat sink is cooled by convection with a co-efficient of heat transfer equivalent to a 0.1 $\frac{K}{W}$ thermal resistance. For an apples-to-apples comparison, we set the heat transfer co-efficient at the convective boundary of *mobile-regular-50* (top of the die) to be such that the peak temperature on the die matches that of *desktop-regular-50* when the power density on the entire die is uniform. The thermal conductivities of the materials are as in section 3.3. The uniform power density on the cores is set to be 1 $\frac{W}{mm^2}$, while that on the L2 cache banks is set to be 0.1 $\frac{W}{mm^2}$. In both the desktop and mobile configurations, we also explore the case of infinite cores as a limit study. When the number of cores is infinity, the power density on the entire die becomes uniform with a value equal to the average of the power density on the cores and that on the cache banks, weighted by their respective area occupancies (50% in this case).



(a) Effect of Package - Bode Plot

(b) Impact of die thickness

(c) Effect of core orientation and occupancy (desktop)

(d) Effect of core orientation and occupancy (mobile)

Figure 5. Results of the checkerboard experiments. Note the overlap of the curves *rotated-25 and regular-25* **(***rotated-75/regular-75***) in (c) and (d).** *alt-25 and regular-25* **are also coincident in (c).**

A steady state thermal analysis of this setup is performed using ANSYS 11.0 FEM solver. For *mobile-*

regular-50, we use a lateral grid size of 96 x 96 on the die and for *desktop-regular-50*, a grid size of 48 x 48 is employed (due to software license restriction on the number of nodes). The lateral dimensions of the grid cells in the other layers are the same as in silicon. The vertical modeling resolution is identical to that in section 3.3. The difference between the peak temperature on the die and the ambient temperature for this study is plotted in figure 5(a). Both the axes are logarithmic (except for the infinity point on the x-axis). Since the number of cores is inversely related to the size of a core, it is similar in sense to spatial frequency. Hence, with the number of cores on the x-axis, the graph is analogous to a Bode plot. The low-pass filter behaviour is clearly evident from the graph of *mobile-regular-50* with a cut-off of around tens of cores. On the other hand, the response is quite shallow for *desktop-regular-50.* Actually, in comparison to *mobile-regular-50*, the response has actually "shifted left" in *desktop-regular-50.* The total vertical thickness of *desktop-regular-50* (in silicon equivalents) is much greater than that of *mobile-regular-50* because of the additional layers. Hence, for a given core size in silicon, the *normalized* size (the ratio of the size to the vertical thickness) is much smaller for the desktop configuration than for the mobile configuration. In comparison to two cores, at infinite cores, there is a 14% reduction in peak temperature for *desktop-regular-50*, which translates into an allowable Thermal Design Power (TDP) increase of an equal amount. At 200 cores, this benefit drops to 10%. These results are similar to the numbers reported in [9] Although significant, this is not as good as the benefit for *mobile-regular-50*, which is 45% going from 2 cores to infinite cores and 33.5% at 200 cores. This is because of the heat spreading in copper which smoothes out the response of the low-pass filter.

From the discussion in the previous section, we know that vertical thickness is an important factor to be considered in a study of spatial filtering. Hence, we explore the effect of the die thickness in both desktop and mobile configurations. Figure 5(b) displays the results of such an investigation. It shows the peak temperature for a checkerboard similar to figure 4(a) for die thicknesses of $50\mu m$, $150\mu m$ and $300\mu m$. The experimental setup is similar to the one in figure 5(a) except that the power density on the L2 caches is assumed to be zero. It is evident that die thickness has a great impact on the mobile configuration and a very small effect on the desktop case. This is expected because the die forms the entire heat conduction stack for the mobile case while it is only a small fraction of the total vertical thickness in the desktop case. Also, the trends in spatial filtering (steep response for mobile and shallow response for desktop) remain the same across different die thicknesses. An interesting behaviour in the mobile configuration is that a thinner die does not necessarily mean lower peak temperature. In fact, it turns out to be the opposite for most core sizes. This is because, the thicker the die, the greater is the lateral heat spreading. In the absence of a heat spreader, for a checkerboard arrangement, the reduction in temperature due to additional spreading in silicon (because of a thicker die) is greater than the increase in temperature due to higher vertical thermal resistance. The exceptions to this rule are the endpoints (2 cores and infinite cores), although for different reasons. In the two-core case, the effect of lateral spreading is significantly diminished because all the cores occupy the corners of the die where spreading is possible in only two directions as opposed to four. In case of infinite cores, lateral spreading is made *irrelevant* because of the uniform power density on the entire die. When neighbours have the same temperature, there is no heat flow between them even if the opportunity is offered by the greater die thickness.

An assumption in the previous checkerboard experiments was that the cores occupy 50% of the total die area. In reality, core-to-cache ratio varies across different microarchitectures. Hence, we study the effect of die occupancy by cores here. When the die occupancy is different from 50%, alternative checkerboard configurations are possible. Figure 4 illustrates a few such configurations for an occupancy of 25%. *regular-25* is a derivative of *regular-50. alt-25* is very similar to *regular-25* except that the cores do not occupy the corners of the chip and the cache banks of a particular processor are placed on both sides of the core. This alternative floorplan is examined in relation to *regular-25* for studying the effect of chip boundaries. For a die occupancy of 25%, since the cores shown in figure 4 have an aspect ratio other than one (1:2 in this case), we also study the effect of rotating the cores by 90° so as to nullify the effect of orientation. This arrangement shown in figure 4(d) is called *rotated-25*. We also explore a die occupancy of 75% through arrangements similar to the 25% case. By the same convention, *regular-75* is nothing but the complement of *regular-25* where cores become cache banks and *vice versa*. *rotated-75* is the complement of *rotated-25*. We do not examine an *alt-75* configuration since that would mean sub-dividing a core into two portions and placing them on either side of a cache bank, which would be difficult in practice. The peak temperature for these arrangements in the desktop and the mobile configurations are presented in figures 5(c) and 5(d) respectively.

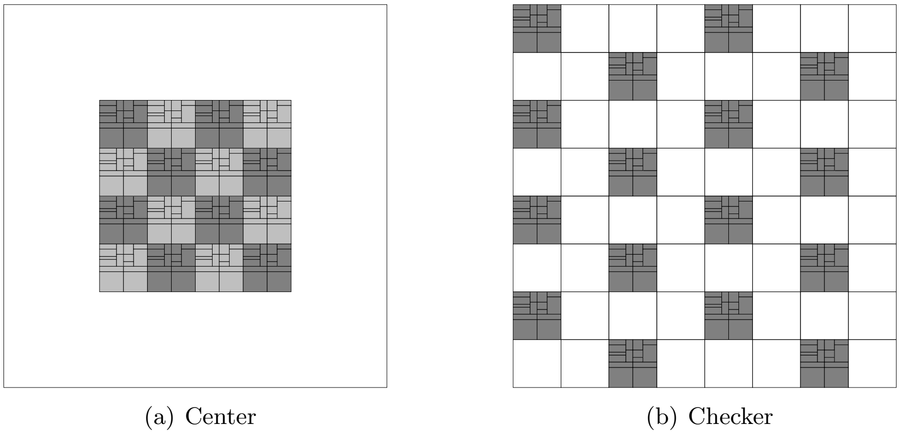
It can be observed that the die occupancy plays the most important role in determining the peak temperature. When the die occupancy is lower, the spatial "duty cycle" is small *i.e.,* two small high power density areas (cores) are separated by a large low power density area (cache banks). On the other hand, the duty cycle is higher for higher die occupancy. Moreover, in comparison with a low occupancy case, the total power dissipated in the corresponding high occupancy die is greater. This is the reason for at least three distinct lines (one for each occupancy level) in each of the graphs 5(c) and 5(d). Another interesting phenomenon is the impact of the different floorplan configurations. Clearly, core orientations do not matter at all for the desktop or the mobile configurations as can be seen from the overlapping lines and points of *regular/rotated-25* and *regular/rotated-75*. Similarly, the fact that *alt-25* is practically on top of *regular-25* for desktop indicates that placing cores in the corner of the chip does not matter in the presence of the heat spreader and the heat sink. This is consistent with our findings in section 3.3 about the location of a power dissipator on the die (*center-4layers-spread vs. corner-4layers-spread*). This is again because of the lateral spreading in the copper layers. While core orientation did not matter for the mobile configuration, there is a clear difference in moving the cores away from the corners of the die. This is the reason the curve for *alt-25* is significantly lower than *regular-25* in figure 5(d). The extra space available for lateral spreading cools down the cores in the corners significantly.

In addition to the above-mentioned parameters, we also vary the magnitude of the cache bank power density and the convection heat transfer coefficient. The effect of these parameters on peak temperature tracks the trends observed above with a steep response in the mobile configuration and a shallow pattern in the desktop configuration. In summary, there is a significant thermal benefit in the choice of many small cores as opposed to a few large cores due to spatial filtering.

4.1.2 Local *vs.* Global Thermal Management

In the context of many-core processors, it is important to study the relationship between core size and the granularity of thermal management. For a given number of cores, it is useful to know the appropriate granularity at which DTM is most effective. Coarser granularity would incur excessive performance cost to contain the temperature below the thermal emergency threshold, while finer granularity would incur superfluous hardware cost. Hence, in this section, we investigate the effectiveness of DTM at various levels of granularity: locally at the level of a functional block (*e.g.*, register file, cache, branch predictor *etc.*), an architectural sub-domain within a core (*e.g.*, the integer pipeline, the floating point pipeline, the memory pipeline *etc.*) or globally at the level of an entire core. This is accomplished through a microarchitectural simulation study of a many-core architecture with each core resembling that of an Alpha 21364 as in [19].

We simulate two different scenarios marking the two extremes of core arrangement with respect to spatial filtering. In the first, all the cores are arranged close to each other at the center of the die with the second level cache surrounding them. The floorplan of such a setup with 16 cores is shown in figure 6(a). The shaded areas denote the cores while the unshaded areas denote the L2 cache (the cores are shaded in


(a) Center

(b) Checker

Figure 6. Illustration of the two different many-core scenarios simulated. The unshaded areas indicate second level cache. The shaded areas denote cores resembling Alpha 21364.

alternating light and dark shades for better visibility). In the second, the cores and the L2 cache blocks are arranged in a checkerboard-like fashion similar to the previous section. The floorplan of such a setup is shown in figure 6(b). The former scenario is called *center* and the latter as *checker*. In both cases, the L2 cache is assumed to occupy 75% of the total die area.

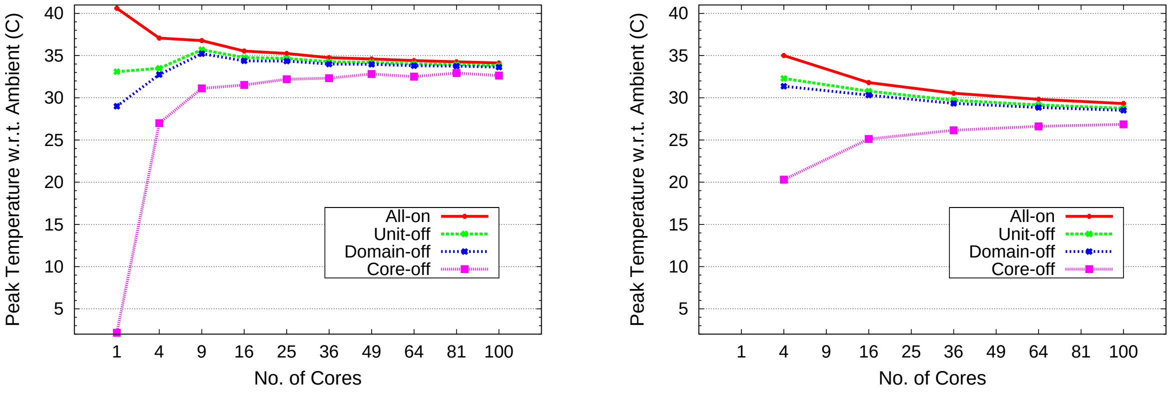
As in the previous section, we vary the number of cores, keeping the power density in each core (and hence the total power on the entire die) constant. In order to examine the effectiveness of DTM, out of all the n cores, we select a single core that is closest to the center of the die and manage its temperature by turning off computational activity in regions of varying sizes. Since we are interested in the best-case DTM potential of a region, we only study the steady state temperature due to its shutdown. If turning a region off does not help in the steady state, it is certainly not going to help in the transient.

The regions are turned off at three levels of granularity. First is at the functional unit level where each of the 15 architectural units are turned off separately, independent of each other. The unit that provides the greatest temperature reduction is chosen as the candidate for comparison against other levels of granularity. The second level of granularity is that of a sub-domain within a core. The core is subdivided into the following four non-overlapping sub-domains:

- *Fetch engine*: I-cache, I-TLB, branch predictor and decode logic.
- *Integer engine*: Issue queue, register file and execution units.
- *FP engine*: Issue queue, register file and execution units.
- *Load-store engine*: Load-store ordering queue, D-cache and D-TLB.

Similar to the functional unit level, each sub-domain is also turned off independent of each other and the best-case peak temperature reduction is chosen. The third and the coarsest level of granularity we consider is that of an entire core. We do not consider coarser levels of granularity (such as groups of cores) here. At the coarsest level, we only turn off a single core closest to the center of the die.

The experimental setup for this study involves the use of SPEC2000 benchmark suite [21] simulated on a tool set involving modified versions of the SimpleScalar performance model [2], Wattch power model [3] and HotSpot 4.1 thermal model [8]. The simulation run of each benchmark involves a subset of 500 Million instructions identified using the SimPoint [18] tool. The average power density values generated from these runs are for a single core at 130 nm. For these steady state thermal simulations, we vary the number of cores by scaling the lateral dimensions of each core linearly. All the cores are assumed to run the same benchmark and hence the power numbers are replicated across them. The thermal model parameters are set to the defaults of HotSpot 4.1 except the die size which is set to 12.4 mm x 12.4 mm (to reflect the 75% L2 cache area) and the convection resistance of the package which is set to $0.75\frac{K}{W}$. We use the grid-based model of HotSpot with a resolution of 256 x 256 grid cells on the die. Of the 26 SPEC2000 benchmarks, only 11 *(7 int (bzip2, crafty, eon, gcc, gzip, perlbmk, vortex) and 4 fp (art, galgel, mesa, sixtrack))* have peak steady state temperatures above the thermal emergency threshold of $85^\circ C$. Since only these benchmarks need DTM, the results presented here are averages over these 11 benchmarks.



(a) Center

(b) Checker

Figure 7. Results of the local *vs.* **global thermal management study.**

Figure 7 presents the results of this study. It plots the difference between the peak and the ambient temperatures for the various levels of DTM granularity as a function of the number of cores averaged over the 11 benchmarks listed above. Figure 7(a) plots the results for the *center* scenario with the cores at the center while figure 7(b) plots the same for the *checker* scenario. The *All-on* curves denote the case with no DTM. It can be seen that it is a decreasing curve with about 16% temperature reduction going from a single core to 100 cores for the *center* case and 16% going from 4 cores to 100 cores for the *checker* case. This temperature reduction is solely due to spatial filtering since the power density and total power remain constant across the number of cores. This is consistent with the findings in the previous section. Also, the curve for *checker* is lower than that for *center* because there is a much wider spatial distribution of heat in the latter because the cores are dispersed.

The *Unit-off*, *Domain-off* and *Core-off* curves denote the best-case (lowest) peak temperatures for the three levels of DTM granularity. Since the power density is kept constant across the number of cores, the total power dissipated in a single core decreases with increasing cores. Hence, the thermal benefit in shutting down a single core also decreases. This is the reason why the *Core-off* curves are mostly increasing. The *Unit-off* and *Domain-off* curves on the other hand, show both increasing and decreasing behaviours. This is a function of two competing factors: on the one hand, the peak temperature (even with all the units turned on) diminishes due to spatial filtering and on the other, the power dissipated (and saved) per unit (or per sub-domain) decreases due to technology scaling. In the curves for the *center* scenario, the latter predominates the former up to 9 cores after which the former takes over.

The main conclusion one can draw from the graphs is that local thermal management (at the functional unit level or at the sub-domain level) ceases to be effective after 9-16 cores. This can be seen from the minimal difference between the *Unit-off* and *Domain-off* curves relative to the *All-on* curve after 16 cores. In fact, this happens even earlier for the *center* scenario at 9 cores. Also, functional unit level management degenerates into sub-domain level management even at 4 cores. It can also be seen that for 50+ cores, even turning off an entire core is not very different from *All-on*. This suggests that one might have to toggle groups of cores at that core size. Another interesting finding is that even though the integer register file is the hottest unit in most of our experiments, it is not the one that gives the most thermal benefit when shut down. This is a function of two factors: first, a higher amount of power is saved in turning off the data cache; and second, other units like the data cache and the integer ALU are hot as well - only marginally cooler than the register file. For these reasons, the load-store engine is the sub-domain that offers the best thermal benefit when shut down. Finally when the entire core is shut down, the hottest units are actually the ones in the periphery of the core and especially the ones adjoining the integer register file of the neighbouring core in the case of *center* scenario.

We also performed sensitivity studies varying the ratio of the L2 cache area and the package characteristics by removing the heat sink and spreader. The results of these studies were not much different from what has been presented above. Thus, to summarize, we can learn from this section that within-core thermal management ceases to be effective beyond tens of cores.

4.2 Sub-blocks with High Aspect Ratio

We saw in section 3.3.2 that the aspect ratio of a block bears an exponential relationship with its peak temperature. Several microarchitectural sub-blocks like register file entries and cache lines are of very high aspect ratios. Their access can be controlled by software either directly (as is the case with registers) or indirectly (as is the case with cache lines). Hence, in this context, it is interesting to examine the question of whether malicious code behaviour can heat up such sub-blocks to undesirable levels by concentrating activity in them. In this section, we approach this question for the data cache from a spatial granularity angle.

Under typical program behaviour, the data array is usually not the hot spot in a cache. This is because, over the time scales at which silicon heats up, the accesses to the data array are usually well-distributed in space. Also, on every cache access, only the line that is addressed dissipates dynamic energy. On the other hand, the periphery is usually the hot spot since the address/data drivers, sense amps, pre-decoders *etc.* dissipate dynamic energy every time the cache is accessed. However, under malicious program behaviour, a single cache line can potentially become the hot spot if the program directs all cache accesses to it. In this section, we perform a microarchitectural study to examine this possibility.

We model a processor similar to the Alpha 21364 as in [19] but scaled to 90 nm. We first collect a representative set of per-unit power consumption values. We do so by simulating the SPEC2000 benchmark suite over a modeling setup similar to that in section 4.1.2. From these simulations, we select the benchmark *bzip2* as the candidate for further exploration and illustration, since it has the highest average data cache temperature.

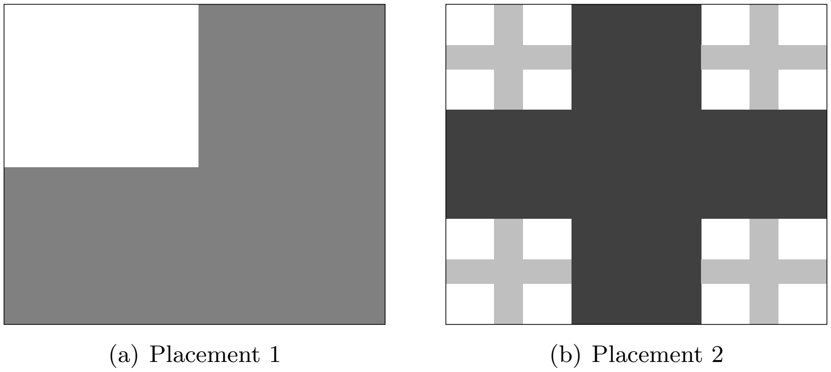
Next, we subdivide the data cache power consumption into those of the individual sub-blocks within the data cache. This is accomplished using the Cacti 5.3 [22] tool that models the performance, dynamic power, leakage power and area of caches. We use Cacti to model an Alpha-like data cache (64 KB, 2-way, 64-byte lines) and separate the data array into the active lines that dissipate dynamic power and the remainder of passive lines that dissipate leakage. We then aggregate the periphery of each sub-array into a single sub-block (including the row and column decoders, sense amps, muxes, comparators *etc.*). The address-input/data-output drivers and the request/reply networks are aggregated into the "outside mat" sub-block. For ease of modeling, the tag and the data portions of the sub-blocks are aggregated into one. Table 1 shows the area, power consumption and power density of each of these sub-blocks for the *bzip2* benchmark. The area and power numbers are expressed as a percentage of the total cache area and power respectively while the power density is represented as a ratio to the average cache power density. It is to be noted that among these sub-blocks, the location and the power consumption of only the active lines (and hence the passive lines as well) can be easily controlled by program behaviour. The other sub-blocks are typically a lot less amenable to program control. Moreover, their thermal behaviour is fairly similar across all accesses, independent of the address/location of the line accessed. Now, it can be seen from table 1 that much of the cache area is occupied by the address/data drivers and request/reply networks. The SRAM array occupies only about a quarter of the cache area with most of it being passive lines at any given time. The power density in these passive lines (due to leakage) is only about a quarter of the average cache power density. On the other hand, the power density at the sub-array periphery is about 24% more than the average. At any given cache access, the power dissipated in the active line is about 9% of the total cache power. For the 64K cache with 64-byte lines, since there are 1024 lines, the area of an active line is about 1/1024 of the SRAM array area. This is a mere 0.03% of the total cache area. Hence, the power density of an active cache line is two orders of magnitude greater than the average. Such high power density however, is not sustained for the time scale at which silicon heats up (tens of thousands of cycles), since the accesses to the cache lines are usually distributed in space. However, malicious code behaviour can concentrate activity in a single line and sustain such a high power density, potentially rising its temperature to undesirable levels. However, the small area of the line is a mitigating factor in the thermal behaviour due to spatial filtering. Furthermore, the high aspect ratio of a cache line (between two and three orders of magnitude greater compared to the cache itself) is also another mitigating factor. The extent to which the spatial filtering balances the high power density is the question we are trying to explore.

Region	Area %	Power %	Power Density Ratio
Outside Mat	56.2	61.4	1.09
Passive Lines	25.7	7.0	0.27
Sub-array Periphery	18.1	22.6	1.24
Active Line	0.03	9.0	358.4

Table 1. Area, power and power density distribution within the data cache for the *bzip2* benchmark.

In order to study the effect of cache layout on its thermal profile, we investigate two different cache arrangements. Figure 8 illustrates them. The first is a naive arrangement with a single sub-array. The shaded portion includes both the "outside mat" sub-block and the sub-array periphery. The second is a placement optimized for performance with the sub-arrays sized in such a manner (using Cacti) that the delays along the two dimensions are balanced. It subdivides the SRAM array into 16 sub-arrays, each with 64 lines. The unshaded areas denote the SRAM cells. The lightly shaded areas denote the periphery of each sub-array and the darkly shaded area denotes the "outside mat" sub-block. It should be noted that the aspect ratio of a single cache line in *Placement 1* is higher than that of a line in *Placement 2*. Since the power densities of the sub-blocks are the same in both the layouts, in considering the two placements, we are actually investigating the impact of sub-block size and aspect ratio on temperature.

In modeling the thermal distribution of these cache arrangements, a challenge arises because of the vastly differing sizes of the sub-blocks. While the other sub-blocks of the cache are comparable in size, the active cache line is different in size and aspect ratio by about three orders of magnitude. Modeling


(a) Placement 1

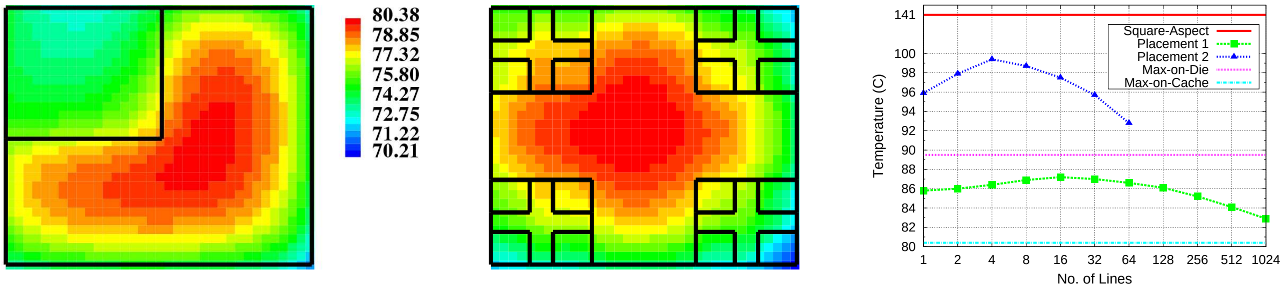
(b) Placement 2

Figure 8. Two different placements studied for the data cache. The unshaded areas denote the SRAM sub-arrays while the shaded portions indicate the periphery and routing.

the entire die at the resolution of the single cache line is prohibitive. In an FEM parlance, doing so would entail millions of FEM nodes. Hence, we perform thermal modeling at two distinct resolutions. First, we model the case with all the sub-blocks except the active line, *i.e.,* all the cache lines are passive, dissipating only leakage power. This is done using the grid-based model of the HotSpot tool at a grid size of 256 x 256 for the entire die (The data cache alone occupies a sub-grid of about 30 x 35). The package parameters are set to their default values except the convection resistance of the package, which is set to $0.75\frac{K}{W}$. Next, we model the active cache line alone (with the rest of the die dissipating zero power). This is done using the ANSYS tool for a die and package configuration identical to the first step above. Due to the small size of the cache line, the thermal distribution of the die in this case is independent of the location of the line within the cache. Hence, we assume that the cache line is at the center of the die and exploit the symmetry of such a setup about the two axes, thereby reducing the number of required nodes by a factor of four. Furthermore, ANSYS employs a non-uniform mesh to model this setup and hence is able to model it with tens of thousands of FEM nodes. Finally, for combining these two steps, since thermal conduction is a linear phenomenon, we use the principle of superposition and sum up the temperatures. It should be noted that since we are interested in the worst-case, we only consider steady-state thermal behaviour here.

Figure 9 shows the results of these experiments. Figures 9(a) and 9(b) plot the spatial temperature distribution of the data cache for the case where all the lines are passive (the first step described above). It can be seen that the hottest region of the cache lies outside the SRAM array in both cases. This is consistent with the power density data since the passive cells have the lowest power density within the cache. It is also clear from the performance-optimized placement that the address/data drivers and the request/reply networks are the most significant contributors to the cache temperature (however, such a conclusion cannot be drawn from the naive placement as it clubs the sub-array periphery and "outside mat" sub-blocks into one).

Figure 9(c) plots the results of including the active cache lines. It plots the peak temperature within the active lines for both the naive and performance-optimized cache placements. Malicious code intent upon heating up the cache can do so either by concentrating activity on a single cache line or on a few contiguous cache lines in a round-robin fashion (to prevent spatial filtering due to the small size and large aspect ratio of a single cache line). In doing so however, it accesses every cache line only once in several (say n) cycles. This reduces the power dissipated and hence the power density by a factor of n. Thus the round-robin technique is actually a trade-off between power density and spatial filtering. Figure 9(c) plots



(a) Placement 1

(b) Placement 2

(c) Active Lines

Figure 9. Results of the data cache thermal experiments. (a) and (b) show the case when all lines are passive. (c) plots the peak temperature of the active lines as a function of the number of lines that are accessed contiguously.

this trade-off for both the cache placements. For the naive placement, with 1024 lines in the sub-array, a round-robin technique can access up to 1024 contiguous lines. Similarly, for the performance-optimized placement, it is restricted to 64 lines since each sub-array only has that many lines. Accessing the lines of another sub-array reduces the advantage of contiguity. The trade-off between power density and spatial filtering is evident in both the *Placement 1* and *Placement 2* curves. They both increase up to a point and then start falling off. The maximum point is when the reduction in power density starts to outweigh the benefit from size and aspect ratio.

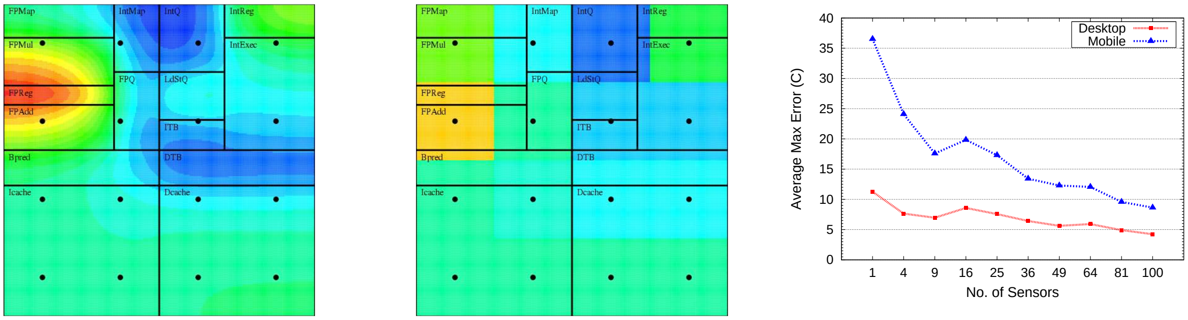
As mentioned above, the thermal behaviour of an active cache line has two mitigating factors that counterbalance its high power density: its size and its aspect ratio. The *Square-Aspect* curve isolates the effect of the former. It shows the peak temperature of a single cache line assuming that it is a square. The area and the total power dissipated are assumed to be the same as a normal cache line. It can be seen from the graph that the *Placement ** curves are far below the *Square-Aspect* curve. This means that aspect ratio is a significant determinant in the peak temperature of a cache line. In fact, this is the reason why the *Placement 2* curve is higher than *Placement 1* (since the aspect ratio of a cache line in *Placement 1* is higher than that of *Placement 2*). Furthermore, the graph also plots the peak temperatures within the cache (*Max-on-Cache*) and across the entire die (*Max-on-Die*) as references. It is to be noted that these reference temperatures are assuming zero active lines.

It can be seen that for *Placement 1*, the steady-state temperature rise due to worst-case code behaviour is only 6.8 degrees above the peak temperature within the cache. This is not sufficient to make the cache lines the hottest spots within the die (as can be seen from *Max-on-Die*). On the other hand, for *Placement 2*, the maximum rise is 19 degrees, which is significant enough to make it 9.9 degrees hotter than the hottest region of the die. As we saw above, this difference is due to the aspect ratio of cache lines in *Placement 1 vs. Placement 2*.

In conclusion, we have provided an example of a cache layout where the application behaviour can cause the lines to become a hot spot and another where this is not possible. However, in both cases, we have shown that aspect ratio plays a crucial role in reducing the peak temperature from an extraordinary value at a square aspect ratio ($141°C$ in our examples) to a much more manageable level ($99.4°C$ and $87.2°C$) in spite of a power density that is orders of magnitude higher than average.

4.3 Thermal Sensors

Thermal sensing is a necessary component of Dynamic Thermal Management (DTM), and sensor accuracy determines the performance of DTM [19]. There can be three major sources of errors in a sensor's thermal measurement. The first is calibration error where the sensor is off by a constant or a linear factor. This can usually be rectified by calibrating the sensor during "burn-in" or if that is not possible, by compensating for it by modifying the temperature at which DTM engages. The second is noise (*e.g.* modeled as following a particular probability distribution like Gaussian, uniform *etc.*). This is usually a temporal effect and the measurements vary around the mean of the distribution. Like calibration error, this noise can also be addressed easily (by averaging across many sample measurements because such error diminishes as the square-root of the number of samples [19]). Since the thermal time constant of silicon is many times greater than the typical thermal sensor bandwidth, multiple readings can be taken before the temperature rises significantly. The third source of sensor error is the spatial thermal gradient. The thermal sensor might not be co-located with a hot spot and hence, the temperature seen by it could be lower than the actual hot spot temperature. This error is the most difficult to address as it depends not only upon the temperature in the immediate vicinity of the sensor but also on its neighbours. It is also directly related to the spatial thermal filtering effect. The steeper the thermal gradient, the greater is this error. Similarly, the farther from the hot spot the sensor is, the greater the error. As Lee *et. al.* [13] point out, there is an exponential relationship between the distance from the hot spot and the sensor error. In other words, similar to the spatial thermal filtering we studied, silicon acts like a low-pass filter for the sensor error due to spatial thermal gradient (with the spatial frequency = 1/inter-sensor distance).



(a) Actual

(b) Sensed

(c) Sensor Error

Figure 10. Relationship between sensor error and inter-sensor distance. For an Alpha 21264-like core, (a) shows a sample temperature profile and (b) shows what is seen by the sensors. The black dots denote the positions of the sensors. (c) plots the average maximum sensor error of the SPEC2000 benchmark suite for two different types of package.

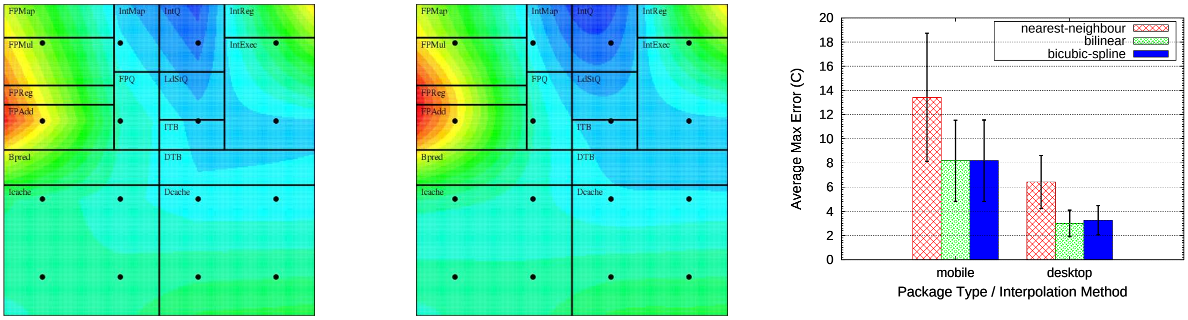
Figure 10 explains this aspect. Figure 10(a) shows the steady state thermal profile of the *sixtrack* benchmark from the SPEC2000 benchmark suite for a 6.2 mm x 6.2 mm die with an Alpha 21264-like core in a "no L2" configuration with a regular grid of 16 thermal sensors marked by the black dots. Figure 10(b) is a picture of what is seen by the thermal sensors. Every point on the die is assigned the temperature of the sensor that is closest to it. This is the so-called "nearest neighbour" algorithm. Figure 10(c) plots the result of an experiment varying the number of sensors and studying the worst-case transient error *i.e.*, the maximum difference between the actual temperature distribution of the die and the thermal profile as seen by the sensors using the nearest neighbour algorithm. The experimental setup is similar to that in section 4.1.2 except that HotSpot 4.1 has been modified to include modeling of a configuration without the package as is the case in hand-held mobile processors. This configuration is denoted by the *mobile* curve. The *desktop* curve includes a typical desktop package with TIM, heat sink and spreader. Furthermore, these experiments are transient thermal simulations with a sampling interval of 0.33 ms. Also, for higher simulation speed, the thermal model resolution is set to a grid size of 64 x 64 as opposed to 256 x 256. At the end of each interval, the maximum error between the actual thermal profile and the interpolated profile across the entire die is computed. We call this error as the *spatial* error. The maximum of these spatial error values is then computed over the whole simulation of 500 million instructions. The maximum error thus computed also includes the *temporal* component of the error. The values plotted in the graph are averages of this maximum error for the entire SPEC2000 benchmark suite.

Clearly, the *desktop* and *mobile* packages behave differently. The former has a shallow curve while the latter has a steep one, similar to what we saw in section 4.1.1. As before, this is due to the better lateral spreading in copper for the *desktop* configuration. Moreover, the sensor error is much higher for the *mobile* curve indicating the need for a higher number of sensors to compensate for the lack of lateral spreading. It can also be seen that the curves show diminishing returns around the mid-range of the curves (mid-tens of sensors) beyond which, increasing the number of sensors does not provide commensurate increase in accuracy. In fact, 36 sensors appears to be an optimal spot in the cost *vs.* accuracy trade-off. Furthermore, the graph shows a non-monotonic pattern. The error is higher for a grid of sixteen sensors than for a grid of nine sensors. This behaviour is because of the location of the sensors relative to the functional blocks. In the increasing portions of the curves shown in figure 10(c), although the grid resolution increases, the sensors move farther from the nearest hot spot, resulting in an increase in the sensor error.

4.3.1 Sensor Interpolation

A natural means to improve the accuracy of temperature estimation using sensors is to employ spatial interpolation in the regions between them. The problem then becomes the estimation of the temperature field on the die at certain desired points (*e.g.* centers of each functional block, borders between two hot units *etc.*), given the n sensor positions and their readings, where n is the number of sensors. Previous research has addressed this problem and solutions involve a heavyweight control theoretic filter (Kalman filter) [17] on the one end of the spectrum and a simple sub-linear interpolator [14] on the other. However, with the insight that spatial filtering behaves differently for different types of package, a spectrum of choices *between* them becomes interesting in the cost *vs.* accuracy trade-off. The objective of this section is to evaluate such a trade-off and characterize a couple of interpolation schemes. Specifically, bilinear and bicubic spline interpolators [16] are two low-cost interpolation schemes that are very popular in the image processing community. In fact, the former is supported in hardware by many Graphic Processing Units (GPU) due to its low computational cost.

For each desired point at which temperature needs to be estimated, the nearest neighbour interpolation algorithm mentioned above is a constant time algorithm. Similarly, the sub-linear interpolator in [14] and the bilinear algorithm are all constant time schemes. At the other end, the steady state Kalman filter is cubic in the number of sensors. An online version of it employed by [17] is quadratic in the number of sensors. The bicubic spline interpolation scheme occupies a position in the middle since it is linear in the number of sensors. It is worth mentioning that while we assume a regular grid of sensors for ease of modeling, the interpolation algorithms themselves are not constrained to a regular grid of thermal sensors. The above-mentioned time complexities of the interpolation schemes assume a regular grid of sensors. In case of non-uniform sensor arrangement, the input to the algorithms would also include the locations of the sensors and an additional step would be required in locating the desired point (where temperature needs to be estimated) relative to the sensors surrounding it. This step would increase the above-mentioned interpolation complexities by an additional $\log n$ term. Furthermore, it should be noted that while Kalman filter is almost a zero-error scheme, its input includes per-unit power consumption estimates which might not be normally available or easily computable. Hence, we do not consider it in our evaluation. Also, the sub-linear interpolation scheme in [14] uses weights ($= 0.5$) such that it is between a zeroth order nearest-neighbour interpolation scheme and a (bi)linear interpolator. Moreover, it employs interpolation only in the vicinity of the hottest sensor, which might be inaccurate if none of the sensors are at the hottest spot. One might have to consider the top k hottest sensors instead of just *the* hottest sensor. Hence, we use the bilinear interpolation scheme as a proxy for the interpolation algorithm in [14].



(a) Bilinear

(b) Bicubic Spline

(c) Results

Figure 11. Results of the interpolation study. (a) and (b) illustrate the behaviour of the bilinear and bicubic spline algorithms for the thermal profile and sensor readings as in figures 10(a) and (b) respectively. (c) presents the experimental results of comparing the algorithms against the nearest-neighbour interpolation scheme.

Figures 11(a) and 11(b) illustrate the operation of the bilinear and the bicubic interpolation schemes for the steady state thermal profile of the *sixtrack* benchmark shown in figure 10(a) and the sensor readings shown in 10(b). It turns out that bilinear interpolation results in discontinuous first derivatives at the sensor locations. Bicubic spline interpolation solves this problem by using a third degree polynomial for curve fit (in each of x and y dimensions) and mandates that the derivatives match at the sensor positions. Hence, it is usually smoother than bilinear interpolation. This can be observed from the pictures, especially near the sensor inside the *FPAdd* unit. It should also be noted that in order to be able to interpolate in the region outside the peripheral sensors but within the chip boundaries, the temperatures of the peripheral sensors are linearly *extrapolated* onto the chip boundaries. The interpolation algorithms are run after this first extrapolation step. Figure 11(c) plots the result of the comparison between the interpolation algorithms for a regular 6 x 6 grid of sensors. The experimental setup is similar to that used for figure 10(c). The graph plots the average maximum error for the entire SPEC2000 benchmark suite with the error bars marking one standard deviation above and below the mean.

The main conclusion is that interpolation reduces the sensor errors significantly (39% for the *mobile* case and 53% for the *desktop* case). However, while interpolation leads to acceptable sensor error in the *desktop* case (three degrees on average), the error is still quite high for the *mobile* case. Hence, the absence of lateral smoothing by copper leads to the necessity for a higher number of sensors even with interpolation. Another observation is that the difference between the performance of bicubic spline interpolation and bilinear interpolation is marginal. In fact, bicubic spline interpolation is even slightly worse than bilinear for the *desktop* case showing that, in this case, a straight line is a better approximation for its thermal distribution than a cubic polynomial. The minimal computational overhead of the bilinear scheme is an added advantage to its accuracy. We also studied the effect of the interpolation schemes for denser sensor grids (up to 10 x 10). The benefit due to interpolation improves with the number of sensors since there are many more points to interpolate from. However, the overall trend remains the same with much of the benefit due to interpolation coming from bilinear interpolation itself.

5 Conclusions and Future Work

This paper presented an analysis of the role of heating granularity on microarchitectural thermal management. It identified spatial thermal filtering as a crucial factor in the study of thermal granularity and derived an analytical equation to model the same. It then explored the agreement of the analytical model with practice and found the behaviour to be dependent on the type of package (*i.e.*, whether a heat spreader and heat sink were present). Using this insight, it then provided three microarchitectural examples where spatial thermal granularity is important:

- It presented a thermal evaluation of a many-core architecture and demonstrated the thermal benefit in choosing many small cores as opposed to a few large cores when all other variables remain constant. It also studied the effect of core size on local *vs.* global thermal management and concluded that local thermal management ceases to be effective beyond low tens of cores.

- It examined the question of whether pathological code behaviour can cause the catastrophic heating of high aspect ratio sub-blocks such as cache lines. While the answer depended on the cache layout and dimensions, aspect ratio always plays a crucial role in mitigating the ills of high power density.

- It explained the relationship between inter-sensor distance and sensor accuracy from a spatial filtering standpoint and studied the impact of bilinear and bicubic spline interpolation techniques on sensor accuracy. Interpolation contributes to the reduction of sensor errors, but the magnitude of the benefit was dependent upon the package. Moreover, the bilinear algorithm was almost as accurate as bicubic spline and more efficient to compute.

In the many-core thermal study, this work identified the ratio of core area to that of the L2 cache to be a crucial determinant of peak temperature. This suggests that a thermally-aware multi-core floorplanning scheme that has flexibility in the use of L2 cache blocks as thermal buffers can potentially reduce the peak temperature significantly. In fact, floorplanning the cores and L2 blocks in such a manner as to minimize the size of the cores and maximize the spacing between them (thereby increasing the spatial frequency to exploit the spatial filtering behaviour) is an interesting possible future direction. This work mainly focused on superscalar cores. With the advent of SIMD architectures like GPUs, the question of whether the different SIMD "lanes" offer a thermal granularity advantage due to their small size is also an interesting area of future work.

References

[1] Ansys 11.0. `http://www.ansys.com`.

[2] T. Austin, E. Larson, and D. Ernst. Simplescalar: An infrastructure for computer system modeling. *IEEE Computer*, 35(4):59–67, Feb. 2002.

[3] D. Brooks, V. Tiwari, and M. Martonosi. Wattch: A framework for architectural-level power analysis and optimizations. In *Proceedings of the 27th Annual ACM/IEEE International Symposium on Computer Architecture*, pages 83–94, June 2000.

[4] H. S. Carslaw and J. C. Jaeger. *Conduction of Heat in Solids: Second Edition.* Oxford University Press, Oxford, UK, 1986, c1959.

[5] P. Dadvar and K. Skadron. Potential thermal security risks. In *Proceedings of the IEEE/ASME Semiconductor Thermal Measurement, Modeling, and Management Symposium (SEMI-THERM)*, Mar. 2005.

[6] R. Dennard, F. Gaensslen, H. Yu, V. Rideout, E. Bassous, and A. LeBlanc. Design of ion implanted mosfet's with very small physical dimensions. *IEEE Journal of Solid-State Circuits*, 9(5):256–268, October 1974.

[7] K. Etessam-Yazdani, H. Hamann, and M. Asheghi. Impact of power granularity on chip thermal modeling. In *ITHERM '06: The Tenth Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronics Systems*, pages 666–670, May/June 2006.

[8] W. Huang, K. Sankaranarayanan, K. Skadron, R. J. Ribando, and M. R. Stan. Accurate, pre-rtl temperature-aware design using a parameterized, geometric thermal model. *IEEE Transactions on Computers*, 57(9):1277–1288, September 2008.

[9] W. Huang, M. R. Stan, K. Sankaranarayanan, R. J. Ribando, and K. Skadron. Many-core design from a thermal perspective. In *DAC '08: Proceedings of the 45th annual conference on Design automation*, pages 746–749, June 2008.

[10] W. Huang, M. R. Stan, K. Skadron, K. Sankaranarayanan, and S. Ghosh. Hotspot: A compact thermal modeling method for CMOS VLSI systems. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 14(5):501–513, May 2006.

[11] E. Kreyszig. *Advanced Engineering Mathematics: 8th Edition.* John Wiley & Sons, Inc., New York, NY, USA, 1998.

[12] J. C. Ku, S. Ozdemir, G. Memik, and Y. Ismail. Thermal management of on-chip caches through power density minimization. In *MICRO 38: Proceedings of the 38th annual IEEE/ACM International Symposium on Microarchitecture*, pages 283–293, November 2005.

[13] K.-J. Lee and K. Skadron. Analytical model for sensor placement on microprocessors. In *Proceedings of the 2005 IEEE International Conference on Computer Design*, pages 24–27, Oct. 2005.

[14] S. O. Memik, R. Mukherjee, M. Ni, and J. Long. Optimizing thermal sensor allocation for microprocessors. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 27(3):516–527, March 2008.

[15] G. E. Myers. *Analytical Methods in Conduction Heat Transfer.* Genium Pub. Corp., Schenectady, NY, USA, 1987.

[16] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical recipes in C (2nd ed.): the art of scientific computing.* Cambridge University Press, New York, NY, USA, 1992.

[17] S. Sharifi, C. Liu, and T. S. Rosing. Accurate temperature estimation for efficient thermal management. In *9th International Symposium on Quality Electronic Design, 2008. ISQED 2008*, pages 137–142, March 2008.

[18] T. Sherwood, E. Perelman, G. Hamerly, and B. Calder. Automatically characterizing large scale program behavior. In *Proceedings of the Tenth International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 45–57, October 2002.

[19] K. Skadron, M. R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan. Temperature-aware microarchitecture. In *Proceedings of the 30th Annual ACM/IEEE International Symposium on Computer Architecture*, pages 2–13, Apr. 2003.

[20] sp3 diamond technologies, diatherm heat spreaders. `http://www.sp3diamondtech.com/prod_diatherm.asp`.

[21] Standard Performance Evaluation Corporation. SPEC CPU2000 Benchmarks. http://www.specbench.org/osg/cpu2000.

[22] S. Thoziyoor, N. Muralimanohar, J. H. Ahn, and N. P. Jouppi. Cacti 5.1. Technical Report HPL-2008-20, HP Laboratories, Palo Alto, April 2008.

[Solution of the Heat Equation]

We re-state the boundary value problem from section 3 (equations (1)-(2)) here.

$$\frac{\partial^2 T_1}{\partial x^2} + \frac{\partial^2 T_1}{\partial y^2} = 0 \qquad (0 \le x \le a, 0 \le y \le l) \tag{6a}$$

$$\frac{\partial^2 T_2}{\partial x^2} + \frac{\partial^2 T_2}{\partial y^2} = 0 \qquad (a \le x \le \infty, 0 \le y \le l) \tag{6b}$$

This is to be solved subject to the boundary conditions:

$$\begin{array}{rcl} T_1(x,l) = 0 & & T_2(x,l) = 0 \qquad (7a) \\ & & T_2(\infty, y) = 0 \qquad (7b) \\ \frac{\partial T_1}{\partial x}|_{x=0} = 0 & & \frac{\partial T_2}{\partial y}|_{y=0} = 0 \qquad (7c) \\ \frac{\partial T_1}{\partial y}|_{y=0} = -\frac{q}{k} & & \qquad (7d) \\ T_1(a,y) & = & T_2(a,y) \qquad (7e) \\ \frac{\partial T_1}{\partial x}|_{x=a} & = & \frac{\partial T_2}{\partial x}|_{x=a} \qquad (7f) \end{array}$$

Of the two parts of the problem, T_2 is the easier one to solve since it is a homogeneous problem (*i.e.*, its boundary conditions are all zero). The standard way to solve for such homogeneous problems is to seek solutions of a *separable* form *i.e.*, assuming $T(x,y) = X(x)Y(y)$. In that case, the steady state heat equation $\nabla^2 T = 0$ reduces to the form $\frac{X''}{X} + \frac{Y''}{Y} = 0$. In this equation, the term on the left side of the + symbol is a function of x alone and the term on the right is a function of y alone. If their sum has to be zero for all x and y, then each term has to separately be equal to a constant (*i.e.*, it cannot be a function of either x or y). Since temperature is a real-valued function, these constants have to be real as well. So, one of them has to be positive and the other negative, so that their sum would be zero. Let us call the positive constant λ^2 (the other would be $-\lambda^2$). Then, depending on our boundary conditions, we typically have two types of equations to solve. The first case is of the form $X'' = \lambda^2 X$ and the second is of the form $X'' = -\lambda^2 X$. The solution to the former is of the form $C_1 e^{\lambda x} + C_2 e^{-\lambda x}$ where C_1 and C_2 are arbitrary constants. This can be seen from the fact that differentiating this expression twice with respect to x results in it being multiplied by λ^2, satisfying $X'' = \lambda^2 X$. Similarly, the solution to the latter case ($X'' = -\lambda^2 X$) is of the form $C_1 \cos(\lambda x) + C_2 \sin(\lambda x)$. This can also be verified to satisfy $X'' = -\lambda^2 X$ as before. Furthermore, since the above-mentioned constants (λ^2 and $-\lambda^2$) occur in positive-negative pairs, it should be noted that when X takes the former form (comprised of exponentials), Y takes the latter form (comprised of sines/cosines) and *vice versa*.

With this introduction to the method of *separation of variables*, let us now apply it to solve for T_2. Let $T_2(x,y) = X_2(x)Y_2(y)$. From the discussion above, $X_2(x)$ can be of the form $C_1 e^{\lambda x} + C_2 e^{-\lambda x}$ or $C_1 \cos(\lambda x) + C_2 \sin(\lambda x)$. Correspondingly, $Y_2(y)$ can be of the form $D_1 \cos(\lambda y) + D_2 \sin(\lambda y)$ or $D_1 e^{\lambda y} + D_2 e^{-\lambda y}$ respectively. However, we know from boundary conditions (7a) and (7b) that T_2 has to be zero at $x = \infty$ and at $y = l$. This restricts X_2 to the exponential form and Y_2 to the sine/cosine form. Thus

$$\begin{array}{rcl} X_2 & = & C_1 e^{\lambda x} + C_2 e^{-\lambda x} \quad \textit{and} \\ Y_2 & = & D_1 \cos(\lambda y) + D_2 \sin(\lambda y) \end{array}$$

Now, applying (7b) to X_2, we get $C_1 = 0$. Also, applying (7c) to Y_2 we get $D_2 = 0$. Therefore, T_2 is of the form $C_2 e^{-\lambda x} D_1 \cos(\lambda y)$. However, when we apply (7a) to T_2, we get $Y_2(l) = 0 \Rightarrow \cos(\lambda l) = 0$.

Since there are an infinite number of λ's that satisfy this equation, we can denote them by λ_n where $n = 0, 1, 2, \ldots \dot{\mathrm{S}}$o, $\lambda_n l = (2n+1)\frac{\pi}{2}$. For each λ_n, we get a different solution. Since the problem we are solving is a homogeneous one, the principle of superposition holds. That is, if there are two solutions satisfying the boundary conditions, their sum is also a valid solution. Hence, all solutions corresponding to the different λ_n's can be added up to obtain T_2. Therefore,

$$T_2 = \sum_{n=0}^{\infty} B_n e^{-\lambda_n x} \cos(\lambda_n y) \tag{8}$$

where B_n's are arbitrary constants. Now, before unraveling T_2 further, let us take a look at T_1. It can be seen that boundary condition (7d) is not homogeneous. In order to be able to apply the method of separation of variables as we did for T_2, we split T_1 into two parts: T_1' and T_p where T_1' is the homogeneous part. T_p is still not homogeneous but is much simpler to construct a *particular solution* for. So, $T_1 = T_1' + T_p$. The original problem $\nabla^2 T_1 = 0$ gets split into two sub-problems:

$$\nabla^2 T_1' = 0, \qquad \nabla^2 T_p = 0 \tag{9}$$

subject to the following boundary conditions:

$$T_1'(x,l) = 0 \qquad T_p(x,l) = 0 \tag{10a}$$

$$\frac{\partial T_1'}{\partial x}|_{x=0} = 0 \qquad \frac{\partial T_p}{\partial x}|_{x=0} = 0 \tag{10b}$$

$$\frac{\partial T_1'}{\partial y}|_{y=0} = 0 \qquad \frac{\partial T_p}{\partial y}|_{y=0} = -\frac{q}{k} \tag{10c}$$

Of the two problems T_1' and T_p, the latter is the simpler one in spite of the non-homogeneity. This is because, unlike T_1' (and the previous case T_2), we are not looking for *all* general solutions. Instead, we are looking for a *particular* solution that suits our problem and boundary conditions. Roughly speaking, we are looking for a function that vanishes when differentiated twice (9) and becomes a constant when differentiated once (10c). Thus, it can be seen that a linear function of y solves the sub-problem (9) by construction, since its second derivative is zero. Hence, $T_p = Py + Q$ where P and Q are arbitrary constants. Applying the boundary conditions (10b) and (10c), we get

$$T_p = \frac{q}{k}(l - y) \tag{11}$$

Now, since T_1' is a homogeneous problem, it can be solved just as we did T_2 by separating the variables, *i.e.*, with $T_1' = X_1(x)Y_1(y)$. Then, by the same arguments as before, one of X_1 and Y_1 must comprise of exponentials and the other of sines/cosines. Applying the boundary conditions (10b) and (10c) to the possible cases narrows down the solution choices to two forms *viz.* $A\cosh(\lambda x)\cos(\lambda y)$ or $A\cos(\lambda x)\cosh(\lambda y)$ where, $\cosh(\theta) = \frac{e^{\theta}+e^{-\theta}}{2}$ and A is an arbitrary constant. Now, for the boundary condition (10a) to hold, the latter choice is not possible because $\cosh(\lambda l)$ can never be zero. Hence, T_1' can only be of the form $A\cosh(\lambda x)\cos(\lambda y)$. Applying (10a) to this $\Rightarrow \cos(\lambda l) = 0$, giving rise to infinite λ_n's as before. So, superposing all such solutions, we get

$$
\begin{aligned}
T_1' &= \sum_{n=0}^{\infty} A_n \cosh(\lambda_n x)\cos(\lambda_n y) && (12)\\
&\textit{(or)} \\
T_1 &= T_1' + T_p \\
&= \frac{q}{k}(l-y) + \sum_{n=0}^{\infty} A_n \cosh(\lambda_n x)\cos(\lambda_n y) && (13)\\
\textit{where, } \lambda_n l &= (2n+1)\frac{\pi}{2} \quad (n = 0, 1, 2, \ldots)
\end{aligned}
$$

From equations (13) and (8), we almost have the complete solutions for T_1 and T_2 except that we need to determine A_n and B_n for $n = 0, 1, 2, \ldots$ This can be done by including the continuity conditions from (7e) and (7f). Substituting for T_1 and T_2 from equations (13) and (8) in the boundary condition (7f) and grouping like terms together, we get

$$B_n = -A_n e^{\lambda_n a} \sinh(\lambda_n a) \tag{14}$$

where, $\sinh(\theta) = \frac{e^{\theta}-e^{-\theta}}{2}$. Now, substituting for B_n from (14) into (8), applying the boundary condition (7e) and grouping like terms together, we get

$$\frac{q}{k}(l-y) = \sum_{n=0}^{\infty} -A_n e^{\lambda_n a} \cos(\lambda_n y) \tag{15}$$

The right side of this equation is an infinite series made of cosines. Hence, it bears a strong resemblance to a Fourier cosine series expansion. Hence, let us expand the function $\frac{q}{k}(l-y)$ using Fourier cosine series and compare with the equation above. In order to be able to expand a function using Fourier series, it has to be periodic. So, let us define a periodic function $f(y)$ which is a periodic extension of $\frac{q}{k}(l-y)$ with a period of $4l$ *i.e.*,

$$f(y) = \begin{cases} -\frac{q}{k}(l-y) & -2l \le y < 0 \\ \frac{q}{k}(l-y) & 0 \le y < 2l \end{cases}$$

and $f(y+4l) = f(y)$. It is to be noted that for our problem, we are only interested in the interval $0 \le y \le l$. Now, we are seeking a Fourier cosine series expansion for $f(y)$ *i.e.*, we are looking for constants a_m ($m = 0, 1, 2, \ldots$) such that $f(y) = a_0 + \sum_{m=1}^{\infty} a_m \cos(\frac{m\pi y}{2l})$. From standard texts on Engineering Mathematics (for *e.g.*, [11]), these can be computed as below:

$$
\begin{aligned}
a_0 &= \frac{1}{2l}\int_0^{2l} \frac{q}{k}(l-y)dy = 0 \\
a_m &= \frac{1}{l}\int_0^{2l} \frac{q}{k}(l-y)\cos\left(\frac{m\pi y}{2l}\right) dy
\end{aligned}
$$

$$
\begin{aligned}
&= \frac{1}{l}\left[\frac{q}{k}(l-y)\sin\left(\frac{m\pi y}{2l}\right)\left(\frac{2l}{m\pi}\right)\right]_0^{2l} \\
&- \frac{1}{l}\int_0^{2l}\sin\left(\frac{m\pi y}{2l}\right)\left(\frac{2l}{m\pi}\right)\left(-\frac{q}{k}\right)dy \quad \textit{(intgeration by parts)} \\
&= \frac{ql}{k}\left[\frac{1-\cos(m\pi)}{\left(\frac{m\pi}{2}\right)^2}\right] \quad \textit{for } m = (1,2,3,\ldots)
\end{aligned}
$$

In other words, $f(y) = \frac{q}{k}(l-y)$ can be written as

$$
\begin{aligned}
\frac{q}{k}(l-y) &= \sum_{n=0}^{\infty} 2\frac{ql}{k}\frac{1}{\lambda_n^2}\cos(\lambda_n y) \\
\textit{where, } \lambda_n l &= (2n+1)\frac{\pi}{2} \quad (n = 0,1,2,\ldots)
\end{aligned}
\tag{16}
$$

Comparing this equation with (15), we can determine A_n as

$$
A_n = -2\frac{ql}{k}\frac{e^{-\lambda_n a}}{\lambda_n^2} \tag{17}
$$

Now, if we substitute A_n and B_n from (17) and (14) into (13) and (8) respectively, we have the full solution to our problem. Before we present the full solution, for notational convenience, let us define $\gamma_n = \lambda_n l$ *i.e.*, $\gamma_n = (2n+1)\frac{\pi}{2}$. Then, the solution for the heat equation is given by:

$$
T_1(x,y) = \frac{ql}{k}\left[1 - \frac{y}{l} - 2\sum_{n=0}^{\infty}\frac{e^{-(\gamma_n \frac{a}{l})}}{\gamma_n^2}\cosh(\gamma_n\frac{x}{l})\cos(\gamma_n\frac{y}{l})\right] \tag{18}
$$

$$
T_2(x,y) = \frac{ql}{k}\left[2\sum_{n=0}^{\infty}\frac{\sinh(\gamma_n\frac{a}{l})}{\gamma_n^2}e^{-(\gamma_n \frac{x}{l})}\cos(\gamma_n\frac{y}{l})\right] \tag{19}
$$

$$
\textit{where,} \quad \gamma_n = (2n+1)\frac{\pi}{2} \quad (n = 0,1,2,\ldots)
$$