An evaluation of HotSpot-3.0 block-based temperature model

Damien Fetis
IRISA/INRIA
Campus de Beaulieu
35042 Rennes cedex, France
dfetis@irisa.fr

Pierre Michaud
IRISA/INRIA
Campus de Beaulieu
35042 Rennes cedex, France
pmichaud@irisa.fr

Abstract

Temperature has become an important constraint in modern microprocessors. Research on temperature-aware computer architecture requires a temperature model. Most recent microarchitecture publications dealing with temperature issues are based on HotSpot, a temperature model specifically developed for such studies. However, to our knowledge, there does not exist any independent evaluation of HotSpot apart from publications from HotSpot authors. This study is a series of comparisons between HotSpot block model and two other models : a finite-element model, and an analytical one. We show that space discretization in HotSpot may introduce a significant error, as is the case with classical numerical methods based on space discretization like finite differences and finite elements. Through this study, we hope to draw HotSpot users' attention to the potential risks in using HotSpot blindly.

1. INTRODUCTION

Temperature is an important constraint in current high-performance microprocessors. Recent high-performance processors feature temperature sensors and control the power consumption to keep temperature below a certain limit. Research on future processors and operating systems must take into account the design constraint. Many interesting studies on temperature-aware computer architecture have been published recently, among which [24, 29, 9, 21, 7, 28, 8, 20, 22, 11, 32, 17, 27, 15] (the list is not exhaustive). These papers show that temperature does not concern only people working on processor packaging or low-power circuit techniques. Temperature problems can be tackled also at the level of the microarchitecture and the operating system. A majority of the papers mentioned above are based on *HotSpot*, a temperature model developed specifically for this kind of research [29]. A software is available on the Internet [2]. As the temperature model is the keystone in temperature-aware microarchitecture studies, it is important to use a reliable model. Since its release, HotSpot has been used by several researchers other that the authors of HotSpot. However, to our knowledge, there does not exist an independent evaluation of HotSpot.

The goal of this study is to draw HotSpot users' attention to the potential risks in using HotSpot blindly without questioning its accuracy. We present a series of comparisons between HotSpot and two other temperature solvers, one using finite elements and the other using analytical methods. The study is divided in two main parts : Section 2 studies the steady-state temperature, and Section 3 studies the time-varying temperature. The main conclusions of our study are :

- HotSpot is sensitive to space discretization. Different floorplans modeling the same power density map may give, with HotSpot, different temperature numbers.
- Floorplans with a larger number of blocks seem to be more accurate
- "Naive" floorplans may give an error exceeding 200%.
- HotSpot underestimates the slope of the temperature response for small times.
- HotSpot underestimates the amplitude of time-varying temperature oscillations

1.1 What is HotSpot ?

HotSpot is based on a network of thermal "resistances" and "capacitances". Such analogy between electric circuits and heat conduction is popular among electrical engineers. The concept of thermal resistance is very convenient for solving rigorously one-dimensional steady-state heat conduction problems where one knows isothermal surfaces *a priori* (e.g., right cylinder with adiabatic sides and uniform boundary conditions at both ends). However, in the general case of three-dimensional heat conduction, the electrical analogy is inexact [12]. Thermal networks are used in the processor-packaging community to characterize packages independently of boundary conditions, leading to so-called *compact* models featuring a relatively small number of resistances and capacitances. There are two main approaches for obtaining a compact model : the behavioral approach, and the structural one [25]. With the behavioral approach, thermal resistances and capacitances values are obtained by calibration based on a detailed model (e.g., finite elements) [12]. In this case, thermal resistances and capacitances have no physical meaning, and changing one package parameter requires to recalibrate the network. With the structural approach, resistances and capacitances values are obtained directly from the package geometry and material characteristics (see [25] for a list of references).

HotSpot is a compact structural model, not for characterizing packages but for modeling microprocessor temperature at the level of microarchitectural units. A description of how thermal resistances and capacitances values are derived is given in [31]. The need for a compact model is justified by the need for fast microarchitectural simulations. HotSpot is intended to allow the user to change the parameters [10], e.g., heat-sink characteristics, material properties, processor floorplan etc.

1.2 Methodology

There are several sources of inaccuracy when modeling temperature [13]. Sources of inaccuracy can be classified into three main categories : inaccuracies coming from physical idealizations (e.g., simplified geometry, linearization, etc.), those coming from not knowing exactly the parameters values (e.g., thermal conductivities, interface material thickness, etc.), and those coming from the mathematical solver (e.g., discretization, truncation of infinite sums, etc.).

This study is not concerned with physical idealizations in HotSpot, but with HotSpot as a mathematical solver. That is, we compare HotSpot with other solvers, using the same physical idealizations and parameter values. We used FreeFEM3D (a.k.a. *ff3d*), a general purpose finite-element solver [3]. However, comparing HotSpot with a single solver would not be sufficient to tell which one is correct in case of disagreement (using a valid tool does guarantee that one uses it correctly). Hence we used a second solver, ATMI, that we developed for our research, using physical idealizations that are close to HotSpot [18]. ATMI is based on classical analytical methods that, unlike finite elements, do not rely on space discretization [5, 16]. When ff3d and ATMI agree with each other but disagree with HotSpot, we conclude that it is an inaccuracy of HotSpot. Although we use HotSpot default parameters, there is always the possibility that we use HotSpot incorrectly, e.g., by using a floorplan for which HotSpot is inaccurate. However, as the HotSpot documentation is not clear about what is a "good" floorplan, the inaccuracies exhibited in this study represent situations that a user may encounter when using HotSpot.

In this study, we evaluate the block-based model of HotSpot-3.0.2. Although the number of nodes in HotSpot can be set arbitrarily large in theory, HotSpot is supposed to be a compact model [10], with a relatively "small" number of nodes, i.e., not exceeding a few hundreds. In the remaining of this study, we use HotSpot accordingly. Unless stated otherwise, we use the default physical parameters of HotSpot-3.0.2. By construction, lateral thermal resistances in HotSpot connect the center of a block to its edges. Hence we assume that HotSpot gives the temperature at the center of each block. In HotSpot-3.0.2, the contact between the heat spreader and the heat sink is assumed perfect. This allows to take a single copper layer in ff3d and ATMI. [1]

The ATMI model is depicted on Figure 1, where layer 1 is the silicon layer and layer 2 is the copper layer. The conductance h_1 between the two layers is computed as $h_1 = k_i/d_i$ where k_i is the thermal conductivity of the interface material and d_i is the interface thickness. The conductance h_2 between the copper layer and the ambient medium is computed as $h_2 = 1/(R_{hs}L^2)$, where R_{hs} is the heat sink thermal resistance and L is the heat sink width. Boundary conditions are listed in Table 1, where T_1 and T_2 are the temperatures in layers 1 and 2 respectively, and $q(x, y, t)$ is the surface power density.

The ff3d model is depicted on Figure 2. There are two differences with the ATMI model. First, there are 3 layers instead of 2, with the middle layer being the interface material. The contact between layers is perfect, i.e., temperature is continuous. Second, the silicon layer width equals the chip width, whereas ATMI does not model chip edges. The physical model here is very close to HotSpot.

[1] We checked that enlarging the heat-spreader in HotSpot so that it matches the heat-sink base does not change temperature numbers.

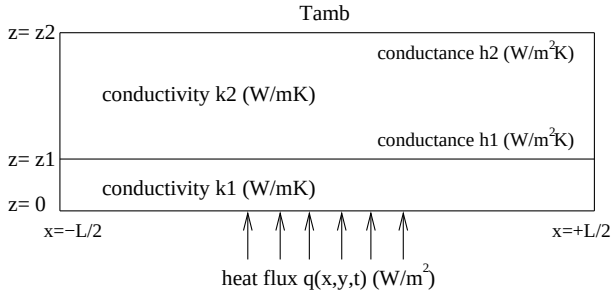


Figure 1: ATMI model : layer 1 ($z \in [0, z_1[$) is silicon, layer 2 ($z \in]z_1, z_2]$) is copper. Heat transfer between the two layers is modeled by a conductance h_1 (W/m^2K). Heat transfer from layer 2 to the ambient is modeled by a conductance h_2. Heat generation is modeled has a prescribed heat flux (W/m^2) on the plane $z = 0$. L is the heat sink width.

locus	boundary condition
$z = 0$	$-k_1 \frac{\partial T_1}{\partial z} = q(x, y, t)$
$z = z_1$	$-k_1 \frac{\partial T_1}{\partial z} = -k_2 \frac{\partial T_2}{\partial z}$ $= h_1(T_1 - T_2)$
$z = z_2$	$-k_2 \frac{\partial T_2}{\partial z} = h_2(T_2 - T_{amb})$
$x = \pm L/2$	$\frac{\partial T_1}{\partial x} = \frac{\partial T_2}{\partial x} = 0$
$y = \pm L/2$	$\frac{\partial T_1}{\partial y} = \frac{\partial T_2}{\partial y} = 0$
$t = 0$	$T_1 = T_2 = T_{amb}$

Table 1: ATMI boundary conditions

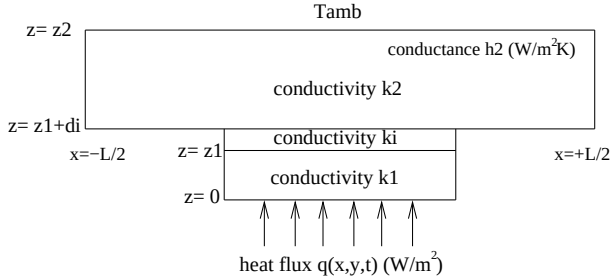


Figure 2: Ff3d model. The contact between layers is perfect. The middle layer is the interface material. The silicon layer width equals the actual chip width.

1.3 Modified HotSpot

HotSpot network features *constriction/spreading* resistances (*constriction* resistances, for short) whose values are obtained with an analytical formula which was derived from [14]. In [14], the formula is obtained for definite boundary conditions, namely a circular right cylinder with Robin conditions on the bottom side and Neumann conditions everywhere else, more precisely adiabatic conditions but in a planar disk source of uniform power density on the top side. Constriction resistances in HotSpot were introduced with the intent to improve the accuracy [30], but the documentation does not mention the fact that the boundary conditions used to derive the constriction resistance formula do not apply. The lack of justification led us to experiment a modified version of HotSpot, where lateral constriction resistances are removed. More precisely, we modified the function *getr* in the file *RCutil.c* so that it returns a simple one-dimensional resistance, namely the value *theta* corresponding to the half-block thermal resistance. In Section 2, we present results both for the original version of HotSpot and for the modified version without constriction resistances.

2. STEADY STATE TEMPERATURE

For the finite-element model in this section, we used ff3d with an external mesh defined with the *gmsh* mesh generator [1]. The mesh has approximately 6.10^5 tetrahedra. For HotSpot, we give results both for the unmodified version (*HS*) and for the modified version without constriction resistances (*HS-mod*).

2.1 EV6 floorplan

In this section, we study different processor floorplans, that are depicted in Figure 3. EV6 is the original HotSpot floorplan. EV6-center is the same floorplan, but with the core at the center of the chip. EV6+ is an enlarged floorplan with $1\,mm$ of non dissipating silicon around. As ATMI does not model the effect of silicon edges, we rely solely on ff3d for quantifying the effect of edges. We expect ATMI and ff3d to give close numbers when silicon edges have little impact on temperature.

Power density numbers are obtained with the *sim-alpha* Simplescalar microarchitecture model [6], and the Wattch power model [4]. We simulate 100 millions instructions of the SPEC2000 benchmark *gzip*, and we provide the steady-state temperature at the center of each unit corresponding to a constant power density equal to the time-average power density in the simulated time interval (this is what HotSpot does, so we do the same with ATMI and ff3d). We provide temperature for two different values of the interface thermal resistance. Default HotSpot values are $d_i = 7.5 \times 10^{-5}\ m$ and $k_i = 1.33\,W/mK$, which corresponds to $h_1 = k_i/d_i \approx 1.77 \times 10^4\ W/m^2K$, that is, a thermal resistance $1/1.77 \approx 0.56\,cm^2K/W$. The second value we used is $h_1 = 10^5\,W/m^2K$, which means taking $k_i = h_1 d_i = 7.5\,W/mK$. The thermal resistance in this case is $0.1\,cm^2K/W$, which is a realistic value [26].

Figure 4 is for the default interface thermal resistance $0.56\,cm^2K/W$. As can be seen, the *Atmi-ev6-center* and *ff3d-ev6-center* temperatures are consistently very close to each other. This shows that the hypothesis of infinitely thin interface in ATMI provides a very good steady-state approximation. This shows also that the chip edges have little impact on temperature when regions of high power density are not too close to the edges. It can be observed that the edges of the silicon die have an impact on temperature for the default floorplan (*ff3d-ev6* vs. *ff3d-ev6-center*). This impact is more pronounced in units whose center is close to an edge, like *IntReg* and *IntExec*. It can be seen that adding $1\,mm$ of non-dissipating silicon

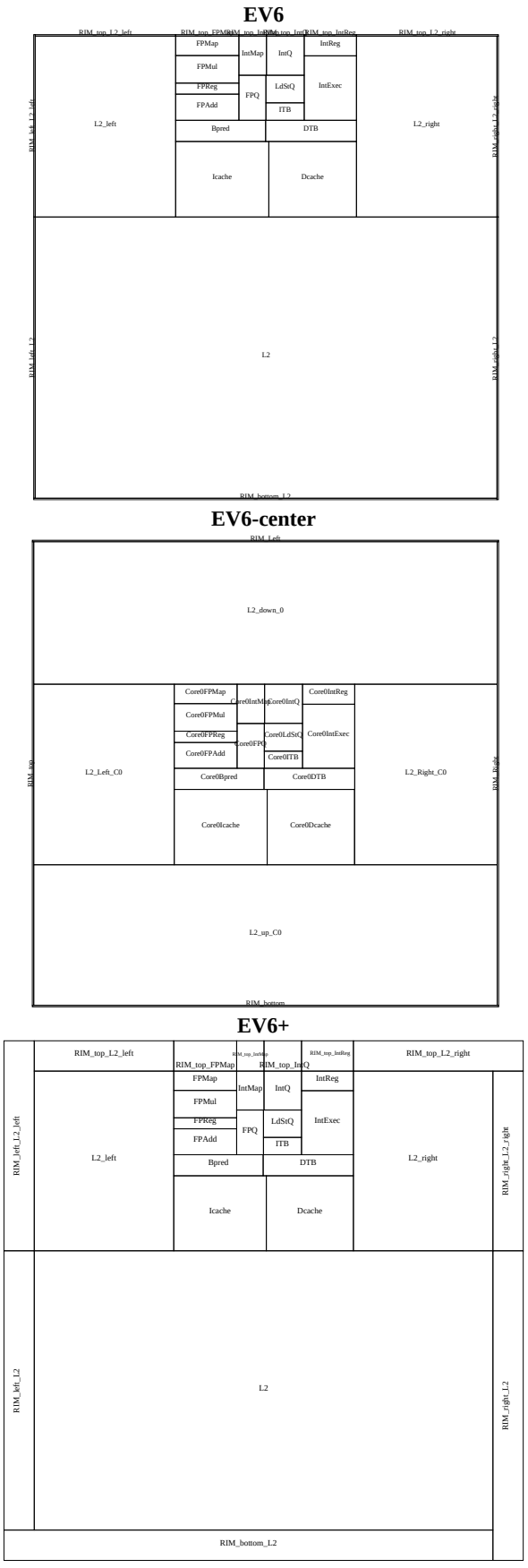


Figure 3: Floorplans : EV6 is the original HotSpot floorplan, EV6-center is the same floorplan with the core at the center of the chip, EV6+ is an enlarged floorplan with $1\,mm$ of non dissipating silicon around.

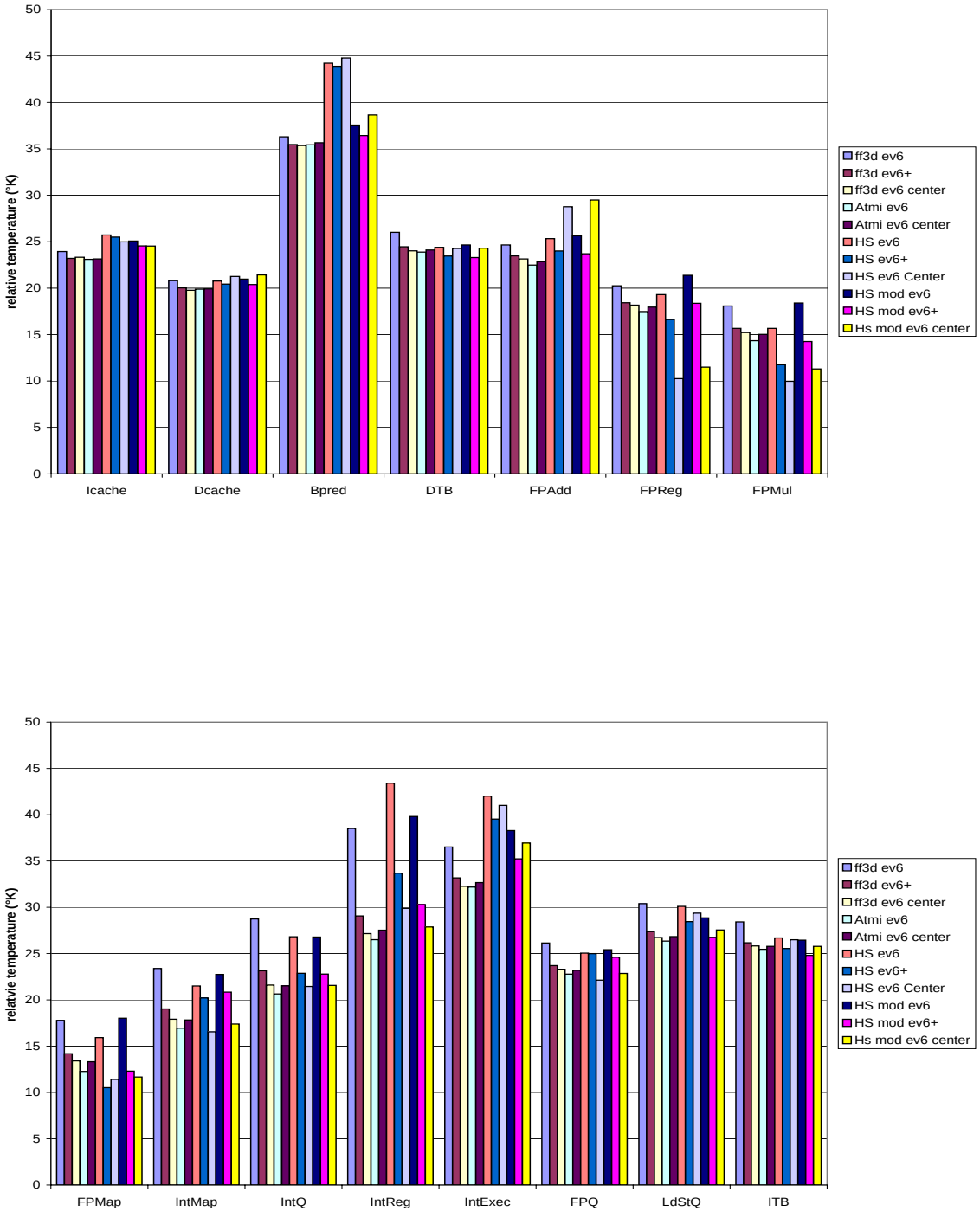


Figure 4: Relative temperature at the center of each EV6 unit for a high interface thermal resistance $0.56\,cm^2K/W$ **(**$h_1 = 17777\,W/m^2K$**).**

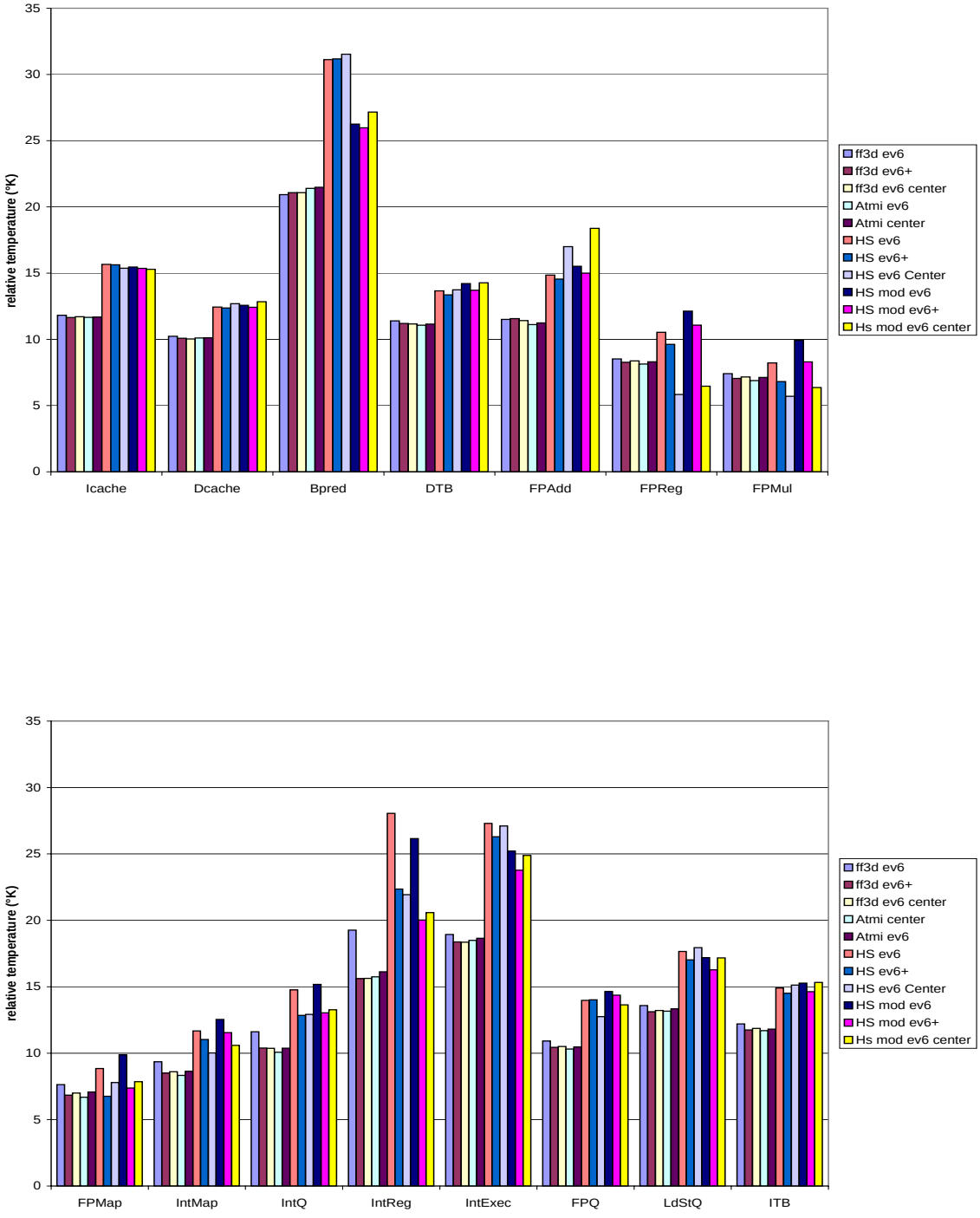


Figure 5: Relative temperature at the center of each EV6 unit for a low interface thermal resistance $0.10\,cm^2K/W$ **(**$h_1 = 10^5\,W/m^2K$**).**

between these units and the die edge permits decreasing substantially the temperature in these units (*ff3d-ev6* vs. *ff3d-ev6+*). This suggests that putting a region of high power density too close to an edge may lead to an artificial temperature problem that could be solved simply by enlarging slightly the silicon die or by changing the floorplan.

As for HotSpot, the comparison between *ff3d-ev6*, *HS-ev6* and *HS-mod-ev6* seems to indicate that HotSpot is more accurate when constriction resistances are removed. Actually, without constriction resistances and using the default parameters and floorplan of the HotSpot-3.0.2 package, the accuracy of HotSpot is relatively good (*ff3d-ev6* vs. *HS-mod-ev6*). Removing constriction resistances is also beneficial on modified floorplans *ev6+* and *ev6-center* (e.g., in *Bpred*). Yet, in some units, the difference between ff3d and HotSpot is significant even without constriction resistances (e.g., in *FPReg*, *ff3d-ev6-center* vs. *HS-mod-ev6-center*).

Figure 5 shows steady-state temperature when taking a low interface thermal resistance $0.10\,cm^2K/W$ ($h_1 = 10^5\,W/m^2K$). As in Figure 4, we see that the *Atmi* and *ff3d-ev6-center* temperatures are very close to each other. However, the impact of silicon edges on temperature is less pronounced when we have a good contact between silicon and copper (*ff3d-ev6* vs. *ff3d-ev6-center*). For example, edges have little impact on the temperature in the *IntExec* unit, which was not the case in Figure 4. Edges still have some impact in the units closest to the edge, like *IntReg*. However, adding $1\,mm$ of non-dissipating silicon permits removing almost completely the edge impact here (*ff3d-ev6* vs. *ff3d-ev6+*). Globally, all temperature have decreased because of the lower interface resistance, and this could explain why the relative difference between ff3d and HotSpot seems larger in Figure 5 than in Figure 4. However, the error is increased not only in relative value but sometimes in absolute value, like in *Bpred*. As previously, removing constriction resistances seems to decrease the error in the hottest units (*ff3d-ev6* vs. *HS-ev6* and *HS-mod-ev6*). However the error is still significant.

HotSpot is sensitive to the floorplan, as can be observed in some units like *FPReg* and *FPAdd* for instance. In these units, the temperature is approximately the same for all three floorplans. Yet, HotSpot gives different temperatures for some floorplans (*HS-ev6* vs. *HS-ev6+* vs. *HS-ev6-center*). These artificial temperature variations are an artifact of HotSpot.

2.2 Square source

In this section, we consider a square source with a uniform power density located at the center of a $21mm \times 21mm$ chip. We give the temperature at the center of the source in function of the source size, for a high ($0.56\,cm^2K/W$) and a low ($0.10\,cm^2K/W$) interface thermal resistance. For HotSpot, we used two different floorplans that are depicted in Figure 6. Though different, these floorplans represent the same power density map, so we should obtain the same temperature.

As can be seen on Figure 7, the 3×3 floorplan (*HS-var* and *HS-mod-var*) is very inaccurate when the central source is smaller than $5\,mm$, with an absolute error of $22\,K$ for a $1\,mm$ source, that is, a 180% relative error. The 3×3 floorplan is most accurate when the source side is $7\,mm$. This corresponds to the case where all 9 blocks have the same size. The 21×21 grid with constriction resistances (*HS-grid*) exhibits a significant error too, though less important than that of the 3×3 floorplan. The smallest error, when the source side is less than $5\,mm$, is obtained on the 21×21 grid without constriction resistances (*HS-mod-grid*).

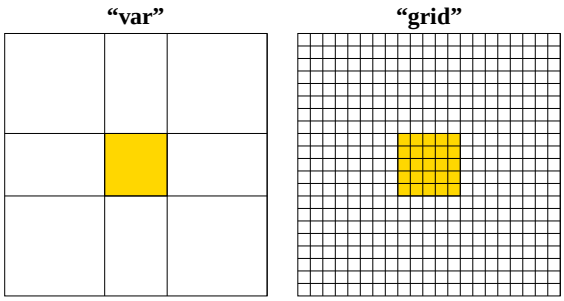


Figure 6: Square source at the center of a $21mm \times 21mm$ chip. The floorplan on the left ("var") defines 3×3 blocks whose size changes with that of the central source. The floorplan on the right ("grid") defines 21×21 square blocks of fixed size.

Figure 8 shows temperature at the center of the source when taking a low interface thermal resistance $0.10\,cm^2K/W$. Compared with Figure 7, the error for small source sizes is bigger here (200% for the 3×3 floorplan). As previously, HotSpot is most accurate on the 21×21 grid without constriction resistances (*HS-mod-grid*). Interestingly, as the source size increases, the error becomes very small. We explain this as follows. As the dimensions of the source approach that of the chip, we become closer to one-dimensional heat conduction case, for which the concept of thermal resistance is accurate. Yet, the reason why the residual error is smaller on Figure 8 than on Figure 7 is not very clear. A possible explanation could be that, by taking a smaller interface resistance, we decrease the temperature gradients on the die plane and make the heat flow closer to a one-dimensional one.

3. TRANSIENT TEMPERATURE

In Section 3.1, we study the temperature response to a step power. Because the system is linear, it can be completely characterized by such temperature responses. In Section 3.2, we study the time-varying temperature using a power trace obtained from simulating a SPEC benchmark. In all remaining experiments, we use the modified version of HotSpot without constriction resistances. [2] We compare HotSpot with ATMI and ff3d in Section 3.1, but only with ATMI in Section 3.2. The reason is as follows. For being accurate with ff3d, we used a fine space discretization (a mesh with 2.5 millions of cuboids). Time must be discretized too, and we used an implicit Euler scheme (a.k.a. backward Euler method). For the temperature response to a step power (Section 3.1), we can increase the time-step progressively, which permits obtaining the temperature curve relatively quickly (hours, not days). However, in Section 3.2, we cannot increase the time-step without hurting accuracy, and the simulation time would be too long.

3.1 Square source

In this section we study the transient response to a step power. We consider, as in Section 2.2, a square source with uniform power density (cf. Figure 6). A power of 10 watts is applied in the square, and we give the temperature at the center of the square in function of time. We consider two different source sizes ($1\,mm$-side and

[2]We checked that removing constriction resistances has little impact on the transient temperature, but improves HotSpot accuracy for large time values, as shown in Section 2.

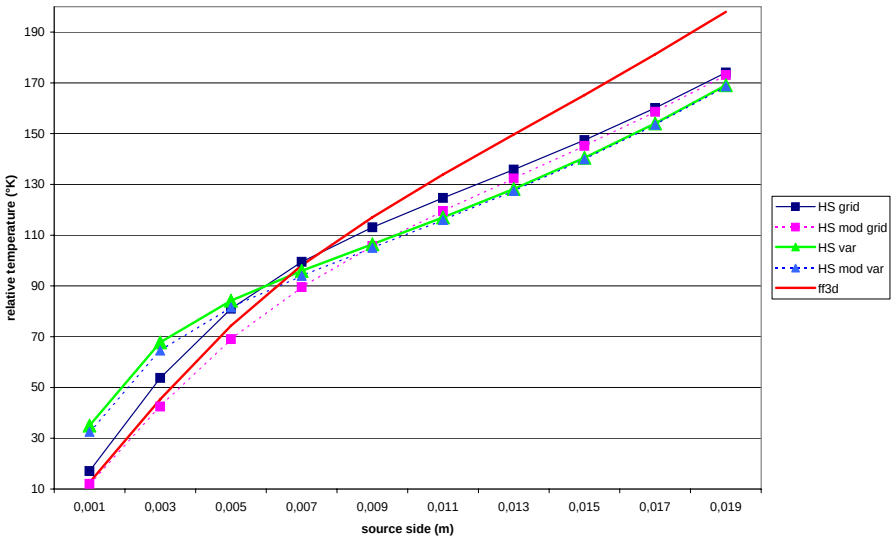


Figure 7: HotSpot and ff3d relative temperature at the center of a square source with (high interface thermal resistance $0.56\,cm^2K/W$). The source power density is $1.66W/mm^2$.

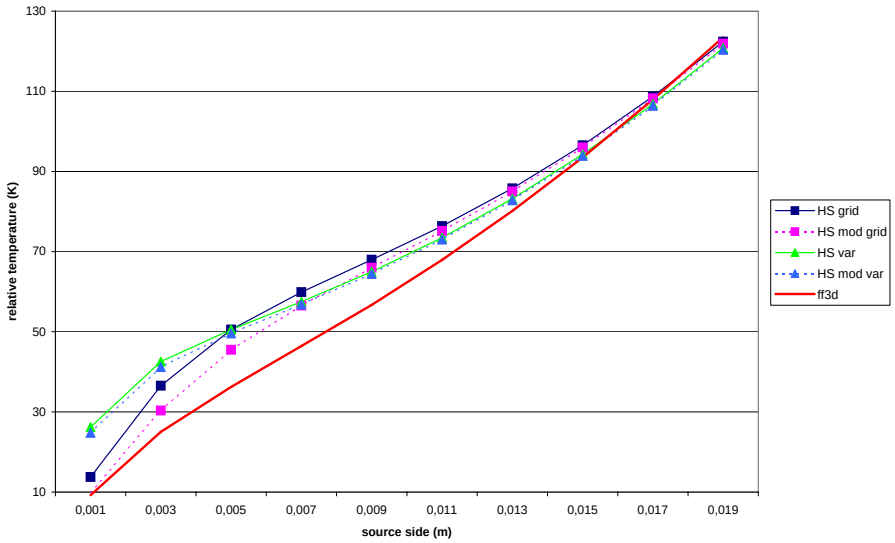


Figure 8: HotSpot and ff3d relative temperature at the center of a square source (low interface thermal resistance $0.10\,cm^2K/W$). The source power density is $1.66W/mm^2$.

$7\,mm$-side) and two different interface thermal resistances (as in Section 2.2 : $0.56\,cm^2K/W$ and $0.1\,cm^2K/W$). For HotSpot, we used the two floorplans depicted in Figure 6. Transient simulation with HotSpot and the 21×21-block floorplan is very slow, so we simulate only the first 0.5 seconds for this case.

Figure 9 is the temperature response for small times and for $1\,mm$-side and $7\,mm$-side sources, with default HotSpot interface thermal resistance ($0.56\,cm^2K/W$). As expected, ff3d and ATMI are close to each other. There is a small difference, mainly due to space discretization in ff3d. HotSpot, on the other hand, exhibits a relatively large error and consistently underestimates temperature for small times. For the $1\,mm$ source, HotSpot behavior can be improved by taking a finer discretization (21×21). However, this improvement is limited because, unlike in ff3d, discretization along the z direction in HotSpot remains unchanged. This is clearly apparent for the $7\,mm$ source, which is closer to a one-dimensional heat conduction problem, and where there is no noticeable difference between the 3×3 and 21×21 floorplans. This inaccuracy of HotSpot, due to a limited space discretization, can be understood by considering the heat equation :

$$k\nabla^2 T + g = \rho c \frac{\partial T}{\partial t}$$

where k is thermal conductivity, ρ the material density, c the specific heat and g the volume power density in W/m^3. At time $t = 0$, temperature is assumed uniform and equal to the ambient, so $\nabla^2 T = 0$, and we have

$$\frac{\partial T}{\partial t}\Big|_{t=0} = \frac{g}{\rho c}$$

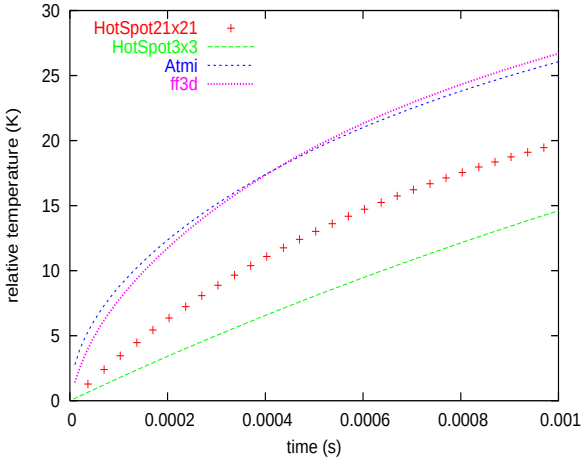
In a processor circuit, heat is generated in a very thin layer, much thinner than bulk silicon. Let us consider a given power P in watts generated uniformly in a layer of thickness e. For a given P, the three-dimensional power density g is inversely proportional to the layer thickness e. In ATMI, the heat dissipation layer is infinitely thin ($e = 0$), and we have

$$\frac{\partial T}{\partial t}\Big|_{t=0} = \infty$$

(actually, for small values of t, the relative temperature is proportional to $\sqrt{t}$). It was shown in [23] that the approximation of infinitely thin dissipation layer is accurate as long as one considers a source whose ratio thickness/width does not exceed $1/20$, which is the case for functional units in typical processors. To reproduce the correct behavior with ff3d, we had to take a fine space discretization. In HotSpot block model, as bulk silicon is modeled with a single network layer, the space discretization is not fine enough. It was already noted in [19] that reproducing the correct behavior with an RC network requires a large number of nodes.

Figure 11 shows the temperature response on a longer time scale. The curves of ATMI and ff3d are close to each other, which shows that the hypothesis of infinitely thin interface in ATMI provides a good approximation in the transient case, as previously observed for the steady state. For a $7mm$ source, we already know that HotSpot steady-state error is high with the low interface resistance value (cf. Figure 8). So it is not surprising that, for large time values, the transient response of HotSpot exhibits a significant error,

$1\,mm$ **source**



$7\,mm$ **source**

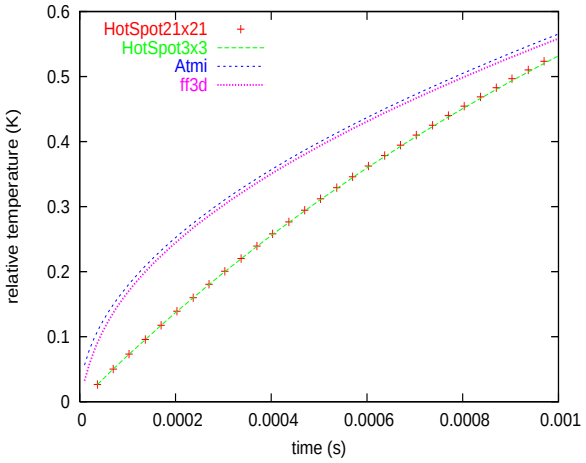
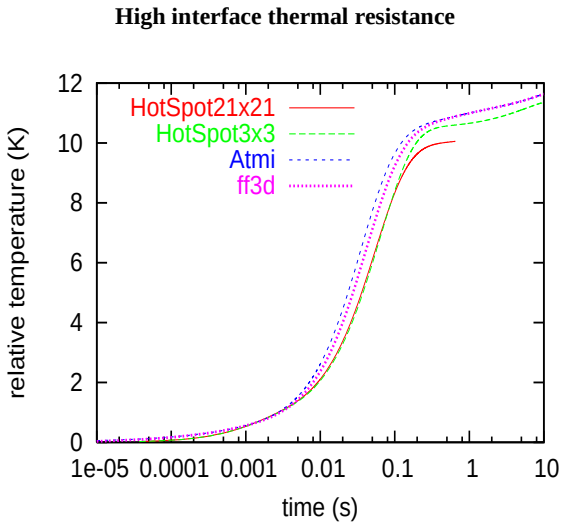


Figure 9: Relative temperature at the center of $1\,mm$-side and $7\,mm$-side square source in function of time (seconds). Time-varying response to a step power, for a high interface thermal resistance $0.56\,cm^2K/W$.



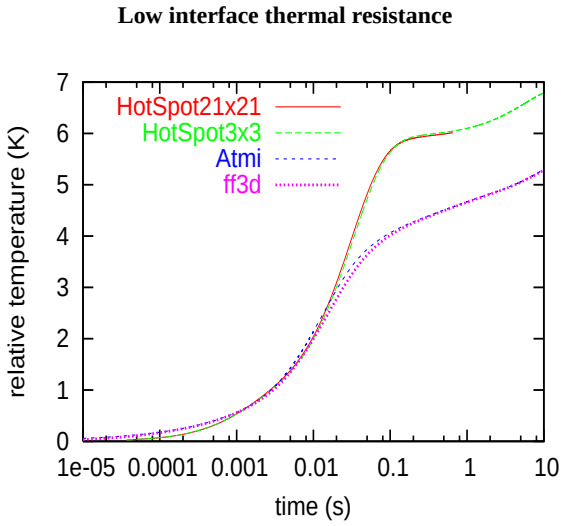
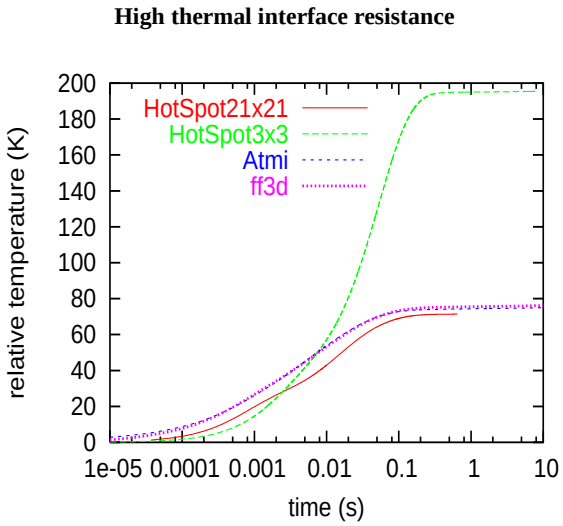


Figure 10: Relative temperature at the center of a $7\,mm$-side square source in function of time (seconds). The upper graph is for a high thermal interface resistance ($0.56\,cm^2K/W$) , the lower graph is for a low interface thermal resistance ($0.10\,cm^2K/W$).



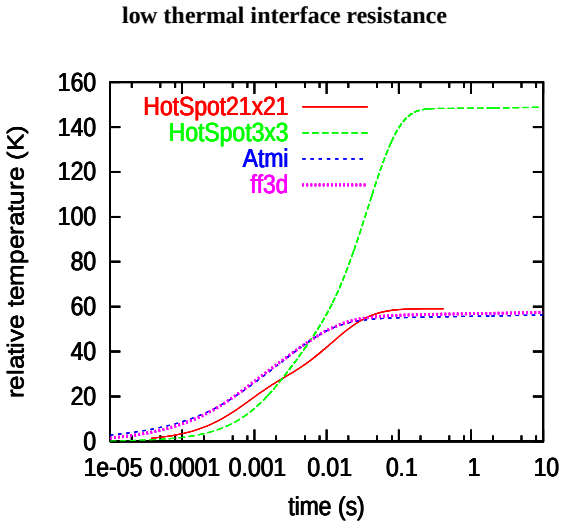


Figure 11: Relative temperature at the center of a $1\,mm$-side square source in function of time (seconds). The upper graph is for a high thermal interface resistance ($0.56\,cm^2K/W$) , the lower graph is for a low interface thermal resistance ($0.10\,cm^2K/W$).

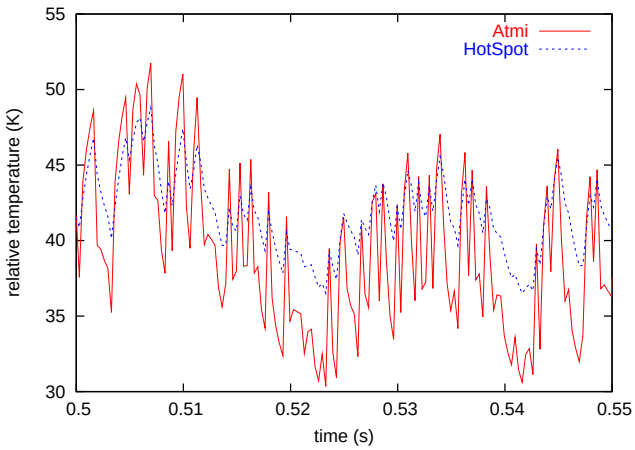


Figure 12: Relative temperature at the center of the branch predictor in function of time (seconds). Response to gzip power trace with *EV6-center* floorplan, for a high interface thermal resistance $0.56\,cm^2K/W$.

as can be seen on the lower graph of Figure 11. For the high interface resistance, the steady-state error is relatively small (cf. Figure 7), and the transient response of HotSpot appears to be not so far from that of ff3d and ATMI. For the $1mm$ source, the steady-state error is high with the 3×3-block floorplan. Figure 10 shows that the error is significant for the transient response too. Taking a finer space discretization (21×21) decreases the error, but it is still significant for small times (and HotSpot is very slow on the 21×21 grid).

3.2 EV6 floorplan

In this section, we study transient temperature with the EV6-center floorplan that we used in Section 2.1. For this study, we use a power trace obtained from *sim-alpha* and Wattch power model. To generate the power trace, we simulate 100 millions instructions with a sampling interval of $100k$ cycles. The clock frequency is 3 GHz, which means a sampling interval of $33\,\mu s$. Simulators are initialized with the steady state temperatures found in Section 2.1.

Figure 12 shows a snapshot of the temperature at the center of the branch predictor in function of time for the *gzip* power trace. The main difference between ATMI and HotSpot is the amplitude of the temperature variations. Temperature oscillations are more pronounced with ATMI than with HotSpot. This observation is consistent with Figure 9, as the small-time temperature response to a step power characterizes the amplitude of high-frequency temperature oscillations.

Not modeling correctly the transients may be misleading, depending on the problem under study. For instance, if one searches the time it takes for a temperature sensor to detect a temperature change of ± 1°C, HotSpot will give a greatly overestimated value.

4. CONCLUSION

HotSpot should be used cautiously. Like finite difference and finite element methods, HotSpot is sensitive to space discretization. However, as HotSpot is supposed to be used with a relatively small number of nodes, there is a risk for the user to obtain inaccurate behaviors. We obtained more accurate results by removing constriction resistances and by using a (relatively) large number of equally-sized square blocks. Yet, as is the case with finite elements and finite differences, a finer discretization in the horizontal plane may not be sufficient, and may necessitate a finer discretization in the vertical direction as well, which HotSpot block-based model, in its current version, does not allow. It is possible that the introduction of a new grid-based model in HotSpot-3.0, by allowing to define several network layers for modeling the silicon die, permits solving the space discretization issue.

Depending on the problem under study, a temperature model with a limited accuracy may be sufficient. Important ideas often resist oversimplifications and are valid for a large range of parameter values (e.g., heat sink and interface thermal resistance values). For instance, in [9], a simplistic temperature model is used to demonstrate the efficiency of activity migration. This does not mean that the qualitative conclusions of [9] are wrong. Subsequent studies based on HotSpot [9], or our own studies based on ATMI [18], confirm the efficiency of activity migration.

In a temperature model for computer architecture research, the accuracy of temperature numbers is inherently limited by the lack of knowledge of parameter values or by approximate power consumption models. What is important is that the model be consistent with physics. A temperature model is also a means for people that are not temperature specialists to gain some qualitative understanding. For example, although HotSpot may provide inaccurate numbers, it exhibits qualitative behaviors, e.g., the fact that the steady-state temperature generated by a heat source decreases as the distance from the source increases, and all qualitative behaviors stemming from the principle of superposition. We believe that many of the intuitions on temperature people can acquire through using HotSpot are correct intuitions.

Nevertheless, HotSpot should not be used without being aware of its limitations and without questioning the physical significance of its outputs. We recall that we have evaluated only the block model of HotSpot, not the newly-introduced grid model. In case one has doubts about HotSpot but still would like to use it, we recommend to use a second tool, e.g. finite-elements, to check that HotSpot, either block or grid model, is correctly calibrated for one's particular use. In case it is not, one may try to remove constrictions resistances or increase the space discretization, as we did.

5. REFERENCES

[1] Gmsh. http://www.geuz.org/gmsh/.

[2] HotSpot. http://lava.cs.virginia.edu/HotSpot/.

[3] freeFEM3D. http://www.freefem.org/ff3d/.

[4] David Brooks, Vivek Tiwari, and Margaret Martonosi. Wattch: a framework for architectural-level power analysis and optimizations. In *ISCA*, pages 83–94, 2000.

[5] H.S. Carslaw and J.C. Jaeger. *Conduction of heat in solids*. Oxford University Press, 1959.

[6] Rajagopalan Desikan, Doug Burger, and Stephen W Keckler. Measuring experimental error in microprocessor simulation. In *Proceedings of the 28th Annual International Symposium on Computer Architecture*, pages 266–277, July 2001.

[7] J. Donald and M. Martonosi. Temperature-aware design issues for SMT and CMP architectures. In *Workshop on Complexity-Effective Design*, 2004.

[8] J. Hasan, A. Jalote, T.N. Vijaykumar, and C.E. Brodley. Heat stroke : power-density-based denial of service in SMT. In *Proceedings of the 11th International Symposium on High-Performance Computer Architecture*, 2005.

[9] S. Heo, K. Barr, and K. Asanović. Reducing power density through activity migration. In *Proceedings of the International Symposium on Low Power Electronics and Design*, 2003.

[10] W. Huang, M.R. Stan, and K. Skadron. Parameterized physical compact thermal modeling. *IEEE Transactions on Components and Packaging Technologies*, 28(4):615–622, December 2005.

[11] J.C. Ku, S. Ozdemir, G. Memik, and Y. Ismail. Thermal management of on-chip caches through power density minimization. In *Proceedings of the 38th Annual International Symposium on Microarchitecture*, 2005.

[12] C. Lasance. Thermal characterization of electronic parts with compact models: interpretation, application, and the need for a paradigm shift. In *Proceedings of the 13th IEEE Semiconductor Thermal Measurement and Management (SEMI-THERM) Symposium*, 1997.

[13] C. Lasance. The conceivable accuracy of experimental and numerical thermal analyses of electronic systems. In *Proceedings of the 17th IEEE Semiconductor Thermal Measurement and Management (SEMI-THERM) Symposium*, 2001.

[14] S. Lee, S. Song, V. Au, and K.P. Moran. Constriction/spreading resistance model for electronic packaging. In *Proceedings of the ASME/JSME Thermal Engineering Conference*, volume 4, 1995.

[15] Z. Lu, J. Lach, M.R. Stan, and K. Skadron. Improved thermal management with reliability banking. *IEEE Micro*, 25(6), November 2005.

[16] D. Maillet, S. André, J.C. Batsale, A. Degiovanni, and C. Moyne. *Thermal quadrupoles - Solving the heat equation through integral transforms*. Wiley, 2000.

[17] A. Merkel, F. Bellosa, and A. Weissel. Event-driven thermal management in SMP systems. In *Second Workshop on Temperature-Aware Computer Systems*, 2005.

[18] P. Michaud, Y. Sazeides, A. Seznec, T. Constantinous, and D. Fetis. An analytical model of temperature in microprocessors. Research report RR-5744, INRIA, November 2005.

[19] Y.J. Min, A.L. Palisoc, and C.C. Lee. Transient thermal study of semiconductor devices. *IEEE Transactions on Components, Hybrids, and Manufacturing Technology*, 13(4):980–988, December 1990.

[20] S.H.K. Narayanan, G. Chen, M. Kandemir, and Y. Xie. Temperature-sensitive loop parallelization for chip multiprocessors. In *Proceedings of the International Conference on Computer Design*, 2005.

[21] M.D. Powell, M. Gomaa, and T.N. Vijaykumar. Heat-and-run: leveraging SMT and CMP to manage power density through the operating system. In *Proceedings of the 11th International Conference on Architectural Support for Programming Languages and Operating Systems*, 2004.

[22] M.D. Powell, E. Schuchman, and T.N. Vijaykumar. Balancing resource utilization to mitigate power density in processor pipelines. In *Proceedings of the 38th Annual International Symposium on Microarchitecture*, 2005.

[23] N. Rinaldi. On the modeling of the transient thermal behavior of semiconductor devices. *IEEE Transactions on Electron Devices*, 48(12):2796–2802, December 2001.

[24] E. Rohou and M.D. Smith. Dynamically managing processor temperature and power. In *Proceedings of the 2nd Workshop on Feedback-Directed Optimization*, 1999.

[25] M.-N. Sabry. High-precision compact-thermal models. *IEEE Transactions on Components and Packaging Technologies*, 28(4):623–629, December 2005.

[26] E.C. Samson, S.V. Machiroutu, J.-Y. Chang, I. Santos, J. Hermerding, A. Dani, R. Prasher, and D.W. Song. Interface material selection and a thermal management technique in second-generation platforms built on Intel Centrino Mobile Technology. *Intel Technology Journal*, 9(1), 2005.

[27] K. Sankaranarayanan, M. Stan, and K. Skadron. A case for thermal-aware floorplanning at the microarchitecture level. *Journal of Instruction-Level Parallelism*, 8, 2005.

[28] A. Shayesteh, E. Kursun, T. Sherwood, S. Sair, and G. Reinman. Reducing the latency and area cost of core swapping through shared helper engines. In *Proceedings of the International Conference on Computer Design*, 2005.

[29] K. Skadron, M.R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan. Temperature-aware microarchitecture. In *Proceedings of the 30th Annual International Symposium on Computer Architecture*, 2003.

[30] K. Skadron, M.R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan. Temperature-aware microarchitecture : extended results and discussion. Technical Report CS-2003-08, University of Virginia, 2003.

[31] K. Skadron, M.R. Stan, K. Sankaranarayanan, W. Huang, S. Velusamy, and D. Tarjan. Temperature-aware microarchitecture : modeling and implementation. *ACM Transactions on Architecture and Code Optimization*, 1(1):94–125, March 2004.

[32] J. Srinivasan and S.V. Adve. The importance of heat-sink modeling for DTM and a correction to ”Predictive DTM for multimedia applications”. In *Fourth annual workshop on Duplicating, Deconstructing, and Debunking*, 2005.