27-29 September 2011, Paris, France

Single-Chip Cloud Computer Thermal Model

MohammadSadegh Sadri, Andrea Bartolini, Luca Benini
University of Bologna
Via Risorgimento, 2, 40136 Bologna, Italy
Tel:0039(0)512093787;Fax:0039(0)512093785, Email:mohammadsadegh.sadr2,a.bartolini,luca.benini@unibo.it

***Abstract*- Spatial and temporal non-uniformities of workload and power consumption advanced Systems-on-Chip (SoC) platforms result in localized high power densities, which lead to temperature hot-spots, gradients and thermal cycles that may cause non-uniform ageing and accelerated chip failure.**

The Single-Chip Cloud Computer (SCC) is an experimental many-core processor created by Intel Labs and it integrates thermal sensors to track the chip thermal behavior. Unfortunately these sensors provide a limited introspection on the full-chip thermal map, as they monitor temperature at a coarse granularity.

In this paper we build a fine-grained thermal and power model of SCC using a state-of-the-art thermal modeling tool (Hotspot). We calibrate the model with measured data from chip sensors. We assess the predictive power of the thermal model and its main sources of error.

I. INTRODUCTION

Upcoming many-cores platforms stress the limits of Moore's law. High clock frequency translates in high power densities that, combined with high spatial and workload variations, produce non-uniform power dissipation which cause non-uniform silicon die temperature. This leads to non-uniform degradation, acceleration of chip aging and increase in cooling costs.

To help designers in studying and addressing these issues at the chip scales, SoC thermal modeling tools [7,8] have been proposed in recent years. HotSpot [2] is a fully parameterized, boundary condition independent, compact thermal model which can be used for exploring the thermal characteristics of ICs at design time as well as when they exist as real hardware. HotSpot provides detailed temperature distribution at different levels such as silicon die layer, package and heat sink surface.

The Single-Chip Cloud Computer (SCC) experimental processor [1] is a 48-core 'concept vehicle' created by Intel Labs as a platform for many-core software research. It has 24 dual-core tiles arranged in a 6x4 mesh. Each core is a P54C core. The SCC die has four on-die memory controllers. It also integrates a small amount of fast local memory located in each tile. Each tile integrates two thermal sensors based on a couple of ring oscillator, one positioned in proximity of the router and the other positioned close to the top core L1 cache. The Board memory controller (BMC) includes a power sensor capable of measuring the full SCC chip power consumption.

We develop a HotSpot thermal model for SCC. The floorplan has been derived directly from SCC specifications [1]. The input power traces of each block composing the floorplan have been extracted from the full chip real power measurements by interpolating the power break-down in [1]. To assess model accuracy and predictive power, we performed a comparison with real measurements under various workloads.

II. SCC MODELLING AND CALIBRATION

A. Floorplan Modeling

To create an accurate thermal model of SCC we generated its floorplan by extrapolating the blocks dimensions and layout directly from design specifications [1]. Fig. 1 shows HotSpot floorplan.

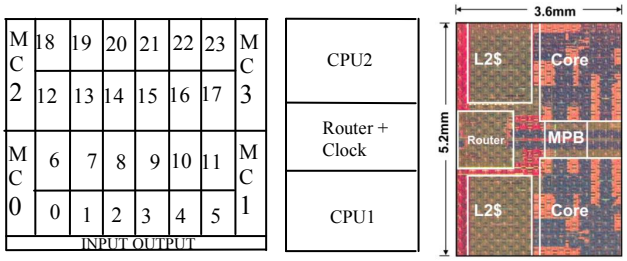

Fig 1. (Left) SCC floorplan for HotSpot (Center) One Tile of SCC in HotSpot floorplan, 2 CPU cores and 1 Router + Clock unit. (Right) One Intel SCC Tile

B. Power Modeling

Accurate thermal simulation requires detailed power traces for each floorplan block. As we can see from the floorplan there are the following basic blocks:

- CPU cores
- DRAM controllers
- Router + Clock
- Serial interface engine (SIF) I/O.

We can obtain the power break-down for these blocks from [1]. In table I, we show the values for two different stress-patterns.

TABLE I
Power break-down of SCC for two sample cases [1]

SCC Unit Name	Full power breakdown (total 125.3W) Cores 1.0GHz. Mesh 2GHz	Low power breakdown (total 24.7W) Cores 125MHz, Mesh 250MHz
CPU Cores	87.7W	5.1W
DRAM Controllers	23.6W	17.2W
Clocking	1.9W	1.2W
Routers	12.1W	1.2W
SIF	~0	~0

From the table we can see that the main contribution to

the total power is given by the CPU and the DRAM controllers.

Unfortunately the power probes available on SCC are not sufficient to probe directly the power consumption of each functional unit. Indeed the available probes are:

- Analog portion of DRAM memory controller
- Digital portion of DRAM memory controller and SIF
- CPU cores, Routers and global clocking

In this section, we first evaluate the sensibility of these probes to different levels of utilization of SCC. We then create a power model suitable to estimate the power consumption of each core given its utilization and workload properties. This will allow us to feed each functional unit in the floorplan with the proper power traces.

The different stress patterns are obtained by executing on different sets of synthetic benchmarks. Each synthetic benchmark is made-up by an infinite loop where an ALU operation is executed on a circular buffer. The dimension of the circular buffer increases within different benchmarks: moving from 16KB to 4MB. At each iteration it is executed a read-write to an entry of circular buffer that moves with an incremental step of one cache line. By doing that, the various synthetic benchmarks are capable of hitting always a data in the L1 cache, missing the L1 cache but hitting the L2 cache and missing L1, L2 and hitting the DRAM. These generate different memory stress patterns.

Table II describes the different workloads properties.

TABLE II
Workloads used for power stressing SCC.
(One cache line of SCC is 32 bytes).

workload	Description
L1	L1 stress: 16KB circular buffer size. Data increment of 32Byte (1 cache line)
L2	L2 stress: 32KB circular buffer size. Address increment of 32Byte (1 cache line)
L2-2Access	L2 stress: 32KB circular buffer size. Address increment of 16Byte (2 access for cache line)
DRAM	DRAM stress: 4MB circular buffer size. Address increment of 32Byte (1 cache line)
DRAM-2Access	DRAM stress: 4MB circular buffer size. Address increment of 16Byte (2 access for cache line)

B. SCC Power Measurement

The first analysis we performed has the goal to highlight how the core power changes in between different cores and workload under stable conditions. To do that we divided SCC into 4 quarters of 12 cores each. Then for each of them we increase the number of cores active. We replicate it for the different synthetic benchmarks. As can be seen, the power increases linearly with the number of active cores for all of the stress workloads. This suggests that the power contribution of the core power is independent on the core position. Thus for the workload used, superposition principle is valid and we can obtain the single core power by dividing the total power to the number of cores.

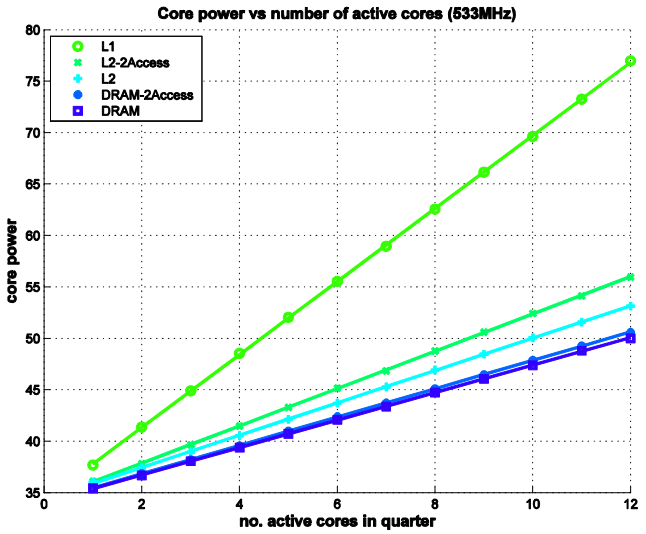

Fig. 2. Core power consumption with number of active cores in each cluster for each of the workloads. Dots are measured data; Lines are the result of linear fitting.

The second test aims to highlight the effect of frequency scaling on different benchmarks. Indeed Fig. 3 shows the cores power consumption when all of them are executing the same benchmark while scaling the tile frequencies.

We can notice that the power changes linearly with the frequency with different slopes for each benchmark[1]. Intuitively this is due to the different CPU-usage of the different synthetic benchmark. A common metric of CPU-load [4] is the Clock Per Instruction (CPI). This can be directly measured at run-time by using the performance counters.

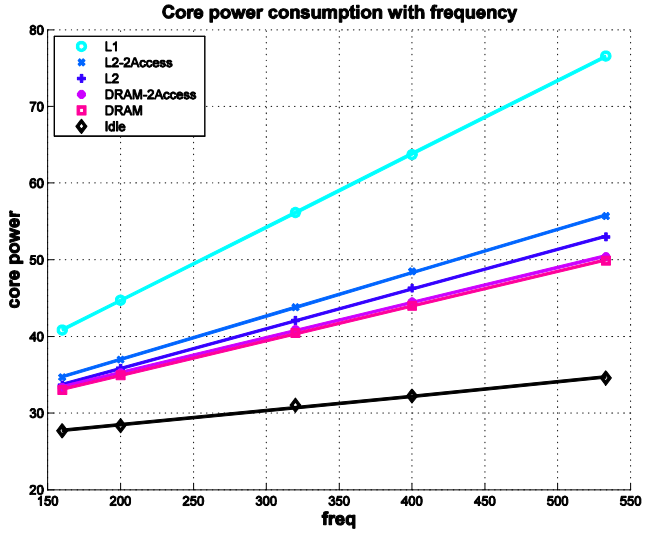

Fig. 3. Core power consumption for each workload for different frequency values. Dots are measured data values. Lines are the result of linear fitting.

Figure 4 shows the CPI of the different benchmarks while changing the tile frequency. We can notice that for L2 and L1 benchmark the CPI does not change with the frequency.

[1] . We scale only frequency because we are not scaling the voltage, to avoid problems on Thermal sensors.

This suggests that both the L1 and L2 are in the same clock domain of the tile. For the DRAM benchmark instead the CPI scales with the frequency and the slope is not constant but changes itself with the frequency.

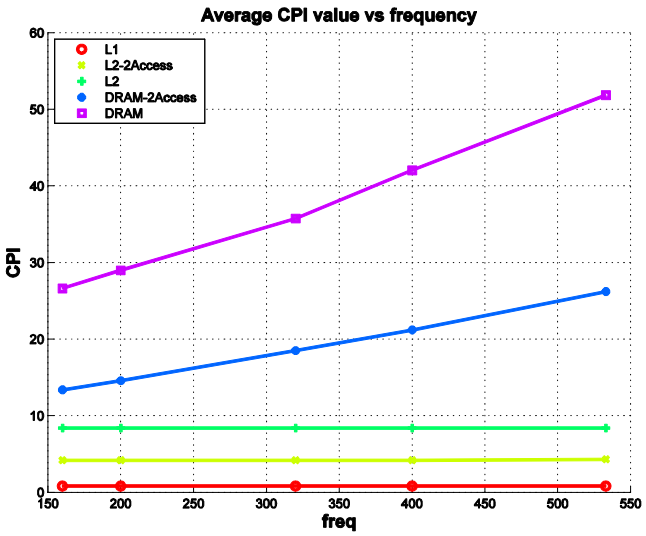

Fig. 4. Changes of CPI for different values of frequency and different workloads.

These effects can be explained by looking at the digital part of DRAM controller power consumption. (Fig. 5) we can notice the same effect. Indeed the power of the DRAM controller shows a saturation effect at higher frequency. This can be explained by bandwidth saturation due to the higher memory accesses issued at higher frequencies.

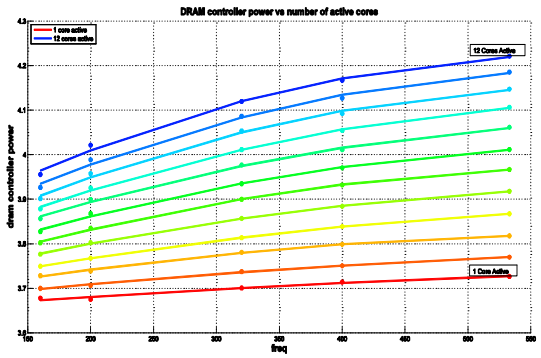

Fig. 5. Power consumption of DRAM controllers as a function of frequency and number of active cores running.

Power Model

As early introduced in this section we combine the core power data values, to derive a power model suitable to predict the core power consumption under different workload and tile frequency.

For each core the power consumption can be split in two main contributions: the idle power and the active power. $P_{core} = P_{active} + P_{idle}$. Whereas the idle one can be modeled as a function of only the frequency ($P_{idle} = g(f_{CORE})$, the active one can be modeled as function of CPI and frequency ($P_{active} = g(CPI_{\#CORE}, f_{TILE})$).

$P_{idle} = b + a.f_{core}$ can be effectively modeled as linear regression of the frequency, whereas P_{active} can be modeled by using equation (1) as the fitting function to generate a closed form model for per-core power.

$$g(CPI, f) = (a + b.CPI^{c}).f + d + \left(a' + b'.CPI^{c'}\right).f \quad (1)$$

In equation (1), z is per-core power value, x is CPI and y is frequency. We use a least square optimization algorithm to find the optimal coefficients that minimizes the error value between estimated and real data. Table III shows coefficients used for computing P_{idle} and P_{active}. Whereas Fig. 6 shows the fitting performance.

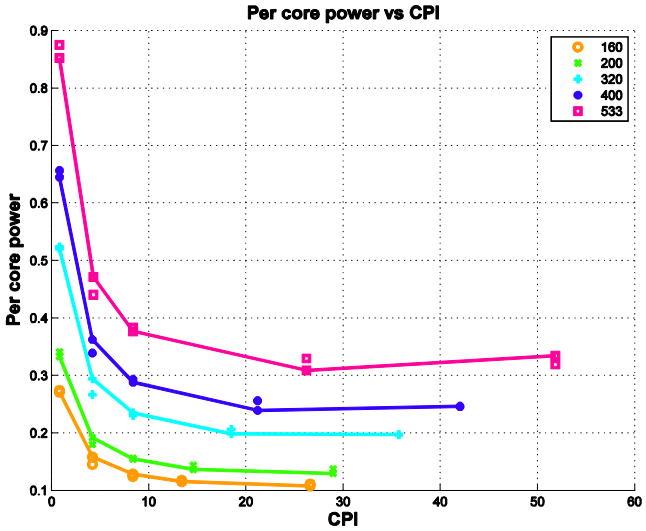

Fig. 6. Power consumption of one CPU core based on CPI values and running frequency. Dots are measured values. Lines are the model output.

Table III
Calculated coefficients for equation 1 to build closed form per-core and per-dram controller power consumption models

P_{idle}	P_{active}
a= 0.35e-3 *b= 0.3872*	*a= 0.0131 , b= -0.240 , c= 0.085* $a' = 0.0131$ *, b'=0.215 , c'=0.02* *d= 0.021*

Since no performance counters are available in SCC to probe on-line the DRAM usage, we decided to use directly the power measurement of the dram controller as input to the thermal model. We feed each memory controller floorplan block in the thermal simulation with 1\4 of the DRAM digital and analog power consumption.

C. *Thermal Sensors Calibration*

As early introduced, SCC integrates in each tile two built-in thermal sensors. The first thermal sensor is placed close to the router and the second one is placed near the L1 cache of the bottom core.

Each thermal sensor is composed of two ring oscillators and the sensor output (TS) is the difference of the two oscillators clock counts over a specific time window *tW*. The difference is proportional to the local die temperature (*T*) [5].

For each tile, the time window (tW) can be programmed through a per-tile control register [6] as a number of tile

clock cycle (NCC) $t_W[s] = NCC/f_{Tile}$. Thus it needs to be updated each time the tile frequency changes.

$$TS = (A + B \cdot T) \cdot t_W \quad (2)$$

where $B<0$ (TS decreases with the temperature rising) and $A>0$ (TS is always positive and in the thousands) are different for each sensors.

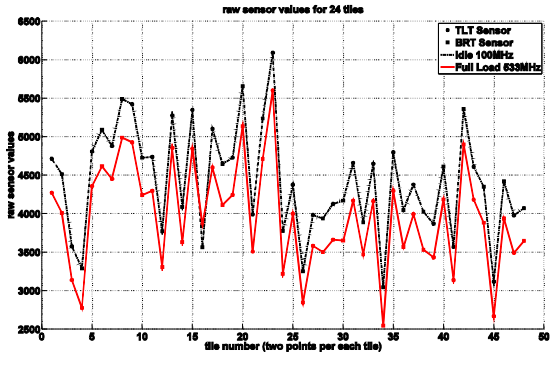

Fig 7.Raw Thermal sensor values

We first evaluate the spatial variation of the thermal sensor values under two different homogeneous stress conditions. Fig.7 shows the results of this test. In the x-axis are reported the thermal sensors moving from the bottom-left corner to the top-right one of SCC (odd ones refer to router sensors whereas even ones to core sensor). The dashed line shows the thermal sensors output when all the cores are in idle state (no task allocated) and are running at the smallest possible clock frequency (100MHz): the coldest point. Instead, the solid line shows the thermal sensors output when all the cores are executing a power virus while running at higher frequency (533MHz). In the coldest point we can assume the real silicon temperature to be roughly constant across the chip area. However we can notice a strong variation in the raw sensor values (>50%). Moreover, it is notable that idle and full-load plots are very similar in shape but for the hotter case (all cores busy) all the sensors output are significantly lower, as expected because sensor count decreases when temperature is high. However it must be noted that the variations due to temperature differences between minimum and maximum load conditions is less than 20% compared to the un-calibrated spatial sensors variation. This clearly highlights the strong need of the thermal sensor calibration step.

We perform a second test with the goal of evaluating the relation between SCC power and thermal sensor output while executing different benchmarks and while running all the tiles together at different frequency levels. Fig.8 shows the results, we can recognize that sensor values are linear with the power consumption.

This property can be exploited to characterize the thermal sensors taking advantage of the linear relation between temperature and power (in steady-state condition) and discovers it by linear regression.

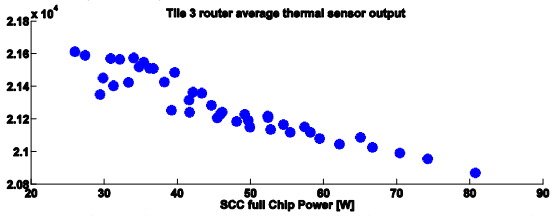

Fig 8. SCC thermal sensors output vs. full chip power consumption

Indeed if we consider the thermal transients expired (steady-state) and we consider a stable workload homogeneously distributed along the multi-core surface (same load/task executing in all of the cores), From equation (2) we can write the thermal sensor output as direct function of full chip power consumption and ambient temperature (which can be measured easily by the end-user)

$$TS_i = A_i + B_i.T_i = A_i + B_i.(K_i.P_{CORE} + T_{AMB}) = A_i + B_i.K_i.P_{CORE} + B_i.T_{AMB} \quad (3)$$

The equation above (3) can be formulated for each thermal sensor (i) as a least square problem where the unknown parameters are A_i,B_i, $B_i.K_i$ and the input data are TS_i, P_{CORE} and T_{AMB}.

Indeed if we generate a cloud of N tuples $\{TS, P, T_{AMB}\}$ by stressing all the cores of the target multi-core together at different frequencies and workloads we can write:

$$\begin{cases} Y_i = \{TS_{i,\#h}\}, h = 1,\cdots,N \\ X_i = \{1, P_{CORE,\#h}, T_{AMB,\#h}\}, h = 1,\cdots,N \\ \Theta_i = \{\alpha_i, \beta_i, \gamma_i\}, \alpha_i = A_i, \beta_i = B_i \cdot K_i, \gamma_i = B_i \\ Y_i = X_i \cdot \Theta_i \Rightarrow \Theta_i = X_i^{\dagger} \cdot Y_i \end{cases} \quad (4)$$

Where N is the number of different stress pattern applied to the system, $X_i^{\dagger}$ is the pseudo-inverse of X_i and Q contains the target calibration coefficients A_i, B_i.

E. Hotspot Thermal Model

We have developed a thermal model for SCC using HotSpot[2] thermal simulator. This model is capable of predicting SCC chip temperature values at different on-die locations according to input power to the chip.

Since there is no public available information on SCC physical process parameters and chip package characteristics we use typical ones as provided in the default HotSpot model.

Hotspot is capable of modeling package heat-sink and forced air convection (using fan). In this case, package convection resistance is computed using the details provided in package model file by the user. This file contains parameters related to dimension of the heat-sink, fan and also rotation speed for the fan. We obtain these by performing measurement of the real physical dimensions of SCC heat-sink: $(H = 8cm, W = 9cm, L = 9cm)$.

For the rotation speed of the fan on SCC platform, there is no direct measurement capability. As consequence we used a typical fan rotation speed for high performance CPU cooling systems (5000rpm).

Table IV shows values of important parameters used for Hotspot simulation.

TABLE IV
Important Hotspot configuration parameter values

Description	Value
Chip thickness	0.15mm
Silicon thermal conductivity	100W/(m-K)
Silicon specific heat	1.75J/(m^3-K)
Heat spreader side	0.03m
Heat spreader thickness	1mm
Heat spreader thermal conductivity	400W(m-k)
Heat spreader specific heat	3.55J(m^3-K)
Interface material thickness	2.0e-2mm
Interface material thermal conductivity	4.0W/(m-K)
Interface material specific heat	4.0e6J/(m^3-K)
Heat-sink fin height	8cm
Heat-sink fin width	1mm
Fan radius	9cm
Fan motor radius	3cm
Fan rotation speed	5000rpm

In order to evaluate the performance of our thermal model, we create a set of test cases. Each test case is composed by:

- A randomly created vector of active cores. This vector indicates for each core, if the core will execute a workload or will be idle.
- A randomly created vector of workloads that will be executed on SCC cores. Each element in the vector specifies one of the available stress workloads. (L1, L2, L2half, DRAM, DRAM-half)
- A randomly created vector of tile frequencies. The items of this vector specify the frequency (between 533MHz to 100MHz) of each SCC tile.

TABLE V.
Comparison of real measurements with model computed data for 10 different tests. (Ambient temperature=313K during the test) (Mean square error is in degrees).

Test NO.	Measure-d Core Power	Estimated Core Power	Mean Square Error	Max error value
1	41.23	40.91	0.59	2.22
2	41.79	41.57	0.31	1.10
3	40.68	40.38	0.46	1.84
4	41.73	41.41	0.53	1.49
5	40.05	39.70	0.55	1.88
6	40.57	40.28	0.61	2.03
7	40.71	40.38	0.55	1.74
8	42.06	41.76	0.47	1.79
9	42.12	41.85	0.44	1.69
10	40.89	40.78	0.60	1.73

We apply each test case to real SCC platform and to our power and thermal model. We then collect the results of each case and compare together. Table V shows the results. The bitmap in the table shows the workload executed on each core and the tile frequency used for each of the cores. The bitmap consists of two rows. Each row contains 48 rectangles. In the top row each rectangle with grey scale, indicates the workload executed on the specific core. The second row instead shows the value of tile frequency. Again the darker the rectangle shows higher frequency.

As we can see in Table V, our power model is capable of estimating SCC core power consumption with a very good accuracy. Even if the max error value is 2 degrees, the average difference between Hotspot estimation and measured values is less than 1 degree.

Fig. 9 shows the SCC thermal map obtained from calibrated sensors with the hotspot simulation under a mixed workload configuration (test case 2). The surface is the output temperature of the thermal model and bars represent the calibrated sensors outputs with a tolerance range (1.5*C*). As we can see, our thermal model is capable of tracking real temperature values under a random workload with good accuracy.

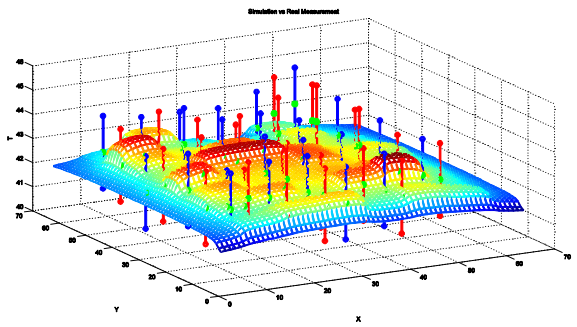

Fig. 9 Comparison of measured temperature values with Hotspot ones when running one of the random test cases.

Figure 10 shows the statistical distribution of error between real results and hotspot one. As can be seen, less than 10% of points have error values larger than 1 degree.

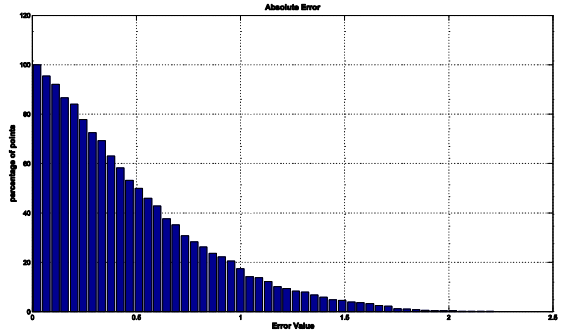

Fig. 10 Statistical distribution of error values. For each error value on x axis, the value on y axis shows the percentage of points with error value higher than the specified error value.

II. CONCLUSION

In this paper, we developed a thermal model for SCC chip. Our focus for SCC power model was on CPU cores. We have presented a calibration approach for the thermal sensors typical for recent and future multi-core architectures. We studied the SCC thermal sensors

performance and behavior. We also performed thermal simulation of SCC using Hotspot tool. We provided a method for tuning the thermal model using the value of convection resistance. We also provided a thermal model considering chip package and heat-sink data. We compared the calibrated sensors values with Hotspot results for some test cases. Comparison shows a tolerance error below 1.5 degrees.

ACKNOWLEDGMENT

This work was supported, in parts, by Intel Corp., Intel Labs Braunschweig and the EU FP7 Projects Pro3D GAn. 248776) and Therminator (GA n. 248603)..

REFERENCES

[1] Howard, J.; and others, "A 48-Core IA-32 message-passing processor with DVFS in 45nm CMOS", *Solid-State Circuits Conference (ISSCC), 2010*

[2] Wei Huang; Stan, M.R.; Skadron, K.;," Parameterized physical compact thermal modeling", *IEEE Transactions on Components and Packaging Technologies, Volume: 28, 2005*

[3] Wei Huang; Ghosh, S.; Skadron, K.;," HotSpot: A Compact Thermal Modeling Methodology for Early-Stage VLSI Design", *IEEE Transactions on Very Large Scale Integration, Volume: 14, 2006*

[4] G. Dhiman, T. Simunic Rosing Dynamic voltage frequency scaling for multi-tasking systems using online learning, ACM International symposium on Low power electronics and design, 2007

[5] Intel Corp., "Using the Sensor Registers", *Revision 1.1,* available at: http://communities.intel.com/community/marc

[6] Intel Labs "SCC External Architecture Specification (EAS)", *Revision 1.1*

[7] Mentor Graphics Corp., "FloTHERM: Optimizing the Thermal Design of Electronics", available at: http://www.mentor.com

[8] Gradient Corp. "HeatWave Thermal simulator", available at: http://www.gradient-da.com/