Temperature-Aware Architecture: Lessons and Opportunities

Wei Huang*, Mircea R. Stan†, Malcolm Allen-Ware*, John B. Carter*, Edmund Cheng§, Kevin Skadron‡

I. Why temperature?

Managing temperature has become one of the most important concerns in modern processor and other microelectronic chips. The problem has become especially severe as the ability to reduce supply voltage has slowed. As a result, the number of devices per unit area is scaling up faster than the power density is scaling down. This requires more expensive cooling solutions in order to keep the chip and its local hot spots cool, and these challenges will be exacerbated by 3D integration, which seems imminent. Furthermore, high temperature makes integrated circuits slower due to degraded carrier mobility and interconnect resistivity. It also accelerates multiple chip failure mechanisms such as electromigration and NBTI, because the wearout rate has an exponential temperature dependency. Static leakage power is primarily an exponential function of temperature. There is also the possibility of thermally-induced security vulnerabilities, such as denial of service [1]. Unfortunately, air cooling's ability to address temperature concerns is limited by system-level power constraints, acoustic challenges, and sometimes also form factors, while alternative cooling solutions are still too expensive for commodity use. Temperature-aware design can reduce these problems.

II. Why temperature-aware, not just power-aware?

To address these thermal concerns requires modeling at every design stage, from early, pre-RTL design exploration to post-layout timing closure. Although temperature is fundamentally a by-product of power consumption, and temperature-aware design is intrinsically related to power-aware design and often uses the same techniques, there are still significant differences between the two design approaches [2]. First, temperature is proportional to power density, not just power. Therefore, in addition to incorporating better cooling solutions, methods to reduce temperature can also either reduce power, or increase area, or distribute power more evenly over a larger area. Low-power techniques could actually increase power density and temperature by using smaller structures and limiting activity to a smaller area. Second, temperature is a non-linear function of time, rising and falling like an RC circuit, while power is an instantaneous value and energy is merely an integral over time. Third, power-aware techniques try to further reduce power consumption when utilization level is low, whereas temperature-aware techniques are primarily engaged when the processor's utilization is high. Fourth, power-aware techniques may create more temperature swings as the chip cycles through high-performance and low-power modes, which harm chip and package reliability, whereas temperature-aware techniques generally try to alleviate these temperature swings. Finally, power-aware techniques usually seek to reduce total chip power and typically are not concerned with localized power densities, while one major goal of temperature-aware techniques is to control local hot spot temperatures.

*Wei Huang, Malcolm Allen-Ware and John Carter are with IBM Research – Austin.

†Mircea Stan is with the Charles L. Brown Department of Electrical and Computer Engineering, University of Virginia.

§Edmund Cheng is with Gradient Design Automation.

‡Kevin Skadron is with the Department of Computer Science, University of Virginia.

III. Why temperature modeling?

During the design of a temperature-aware processor, in order to fully explore the large design space without expensive silicon prototypes, temperature models are needed. One may wonder if power or power density are sufficient as proxies for temperature. Explicit temperature modeling is indeed necessary. Temperature changes gradually in time and space, while power can be a step function. In fact, temperature is a low pass filter, filtering out both high temporal and spatial frequencies. Temperature is also needed to accurately estimate power, because leakage power is exponentially dependent on temperature. None of these phenomena can be inferred without actually modeling temperature and heat transfer. However, it is important to note that temperature is a function of power, so the accuracy and resolution of thermal models is determined by the accuracy and resolution of the power inputs. For example, microarchitecture-level thermal models using power inputs from microarchitecture units cannot be used with any precision for transistor-level temperature estimations.

Figure 1(a) shows components inside a typical server system and the air flows from the inlets all the way to the fan, removing heat generated by different components. A system-level thermal model should model the air flow and account for its impact on the thermal coupling among all the components, such as the impact of hot air flowing off the DIMMs onto the processors. One such model from academia is Mercury [3]. There are also a number of commercial system-level thermal models.

Figure 1(b) shows the typical package components of a modern processor. There are two heat transfer paths. The primary path from silicon to heat spreader and heat sink accounts for about 90% of heat transfer with forced air cooling. The thermal interface material(s) often have more thermal resistance than the other components, making them

essential for accurate modeling. The secondary path from the silicon to C4 pads to package substrate and printed-circuit board becomes dominant with passive cooling. A thermal model for chip architecture design should capture both heat transfer paths with reasonable details in order to achieve good accuracy [4]. There are also a number of thermal models that focus on the chip, accounting for the temperature deltas within the chip given a particular air temperature (e.g. [5], [6]).

IV. Why early temperature-aware design

Now, at which design stage should we start to include temperature considerations with the aid of thermal models? The answer is to include them as early in the design cycle as possible, before making decisions that unwittingly rule out the most effective design choices for managing temperature, or even unwittingly commit to choices with severe thermal consequences. Here we give several examples.

A. Cooling solution changes chip architecture

Li et al. investigate the impact of the cooling solution on multi-core chip architecture [7]. For a system with a high-end cooling solution, thermal constraints are less severe. This generally favors complex cores with more power-hungry and high-power-density structures to exploit instruction-level parallelism, and also allows more cores in a chip. On the other hand, a mediocre thermal solution shifts the optimal configuration towards fewer and simpler cores with narrower issue width and shallower pipelines, as these cores have less severe local hot spots as well as generally consuming less power. The most important observation is that core *type* is an important lever, yet if the wrong type is chosen for detailed design and only later discovered not to match the capabilities of the cooling solution, core *count* and voltage and frequency will be the main remaining ways to compensate, leading to severely sub-optimal performance or dramatically higher cooling costs.

A heterogeneous manycore design with one complex primary core and many simpler cores may suffer from the impact of local hot spots inside the large core, especially if it is "boosted" to run at high frequencies when the other cores are idle. On the other hand, a homogeneous manycore design has more uniform temperature distribution, and hence is more immune to the ability of cooling solutions to smooth out hotspots. Recent work [8] shows that the thermal constraint imposed by a commodity cooling solution makes performance largely insensitive to the complexity of a boostable primary core across diverse degrees of parallelism.

Recently, various studies have explored novel cooling solutions. One example is a heatsink design with localized cooling [9] that provides two different coolant flow paths for hot spot and the rest of the chip, if the hotspots are known in advance. Microchannel cooling (both 2D and 3D) is an example of proximity cooling, where coolant flows through microchannels cut in the silicon substrate and provides a short heat transfer path from the heat sources to the ambient. Huang et al. [10] show that although 2D microchannel can tolerate much higher power densities, increasing chip sizes pose a practical limitation as pumping coolant become much less efficient through longer microchannels [11]. All these advanced cooling solutions would allow significantly higher local power density, and favor tightly clustered cores to reduce communication latency and enable resource sharing. On the other hand, conventional cooling solutions with a long heat transfer path from hot spots to ambient may favor a distributed-core configuration where cores are separated from each other by last level caches that have low power density and can act as thermal buffers.

All these examples show how the cooling solution changes the processor architecture–yet in current practice, the architecture is typically set before the cooling solution is chosen.

B. Cooling power is not free – system architecture

Temperature-aware chip design can also affect system-level design. A thermally optimized processor saves precious cooling power (such as power spent in the fan) by allowing better thermal balancing between processors and other components of the system such as memory DIMMs and disks, as the processors are usually the thermal bottleneck. This is especially beneficial in power-capped high-end servers. For example, recent work by Shin et al. [12] investigated power optimization between cooling fans and processor. The tradeoff here is the cooling fan power and processor leakage power, which is exponentially dependent on temperature.

C. Device modeling

Temperature also matters during synthesis. For example, temperature can affect timing closure: circuit paths in or near hotspots may violate timing constraints, limiting potential operating frequency. As another example, many characteristics of analog/mixed-signal circuits (e.g. the output power of a transceiver circuit in a SiGe BiCMOS design) are extremely sensitive to thermally-induced mismatches, as observed by Gradient DA [13]. Such problems can be identified and repaired with appropriate thermal modeling, before the resulting limitations are committed to silicon.

D. 3D integration

As the industry begins to move toward 3D integration, early-stage temperature-aware design becomes even more crucial, as 3D integration significantly increases power density. As layers from heterogeneous semiconductor processes are integrated into the same package, existing thermal challenges like those above are magnified, and new thermal-related issues arise, such as overlapping hotspots.

V. Challenges

In order to excel in temperature-aware architecture design, industry and academia must address a number of remaining challenges. Here we identify a few.

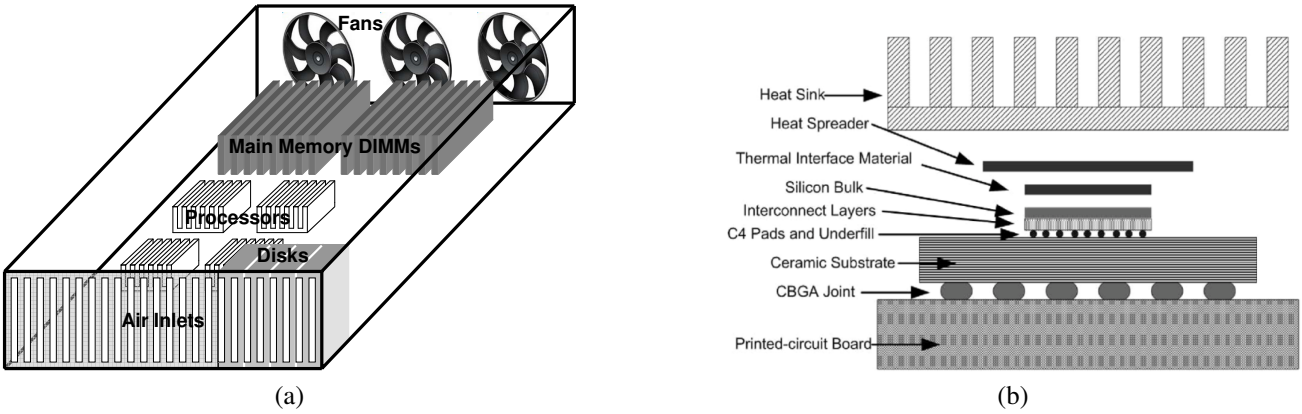

Fig. 1. (a) Major components and a typical cooling solution of a server. (b) Heat transfer paths inside a modern processor package.

A. Dynamic thermal management is hard

If cooling is sufficient, runtime thermal management is only needed as a failsafe to protect against extraordinary programs, extreme environmental conditions, or hardware failure. However, as power density and total power rise and hit cooling limits (whether due to intrinsic limits, form factors, or merely cost), processors may be forced to operate below the circuits' intrinsic performance capability. This requires more sophisticated, efficient runtime thermal management with minimal performance overhead. The challenge, of course, is that thermal management is most needed when the workload is placing the greatest demand on the system and hence is most sensitive to overheads. Dynamic thermal management (DTM) has been extensively studied; a survey can be found in [14]. Most prior work has focused on throttling throughput (often via dynamic voltage and frequency scaling) or migrating between hot and cold resources to achieve a more even spatial distribution of power dissipation. The former approach, which spreads out work in time, inevitably incurs some slowdown. The latter has the potential to avoid slowdown, but spreading out units typically increases communication latencies, and migrating tasks incurs some slowdown as well. DTM is particularly challenging in real-time contexts, where unexpected delays can disrupt real-time schedules.

DTM becomes even more complicated in the presence of manufacturing variations in both silicon and package (e.g. thermal interface material thickness variation [4]). Dynamically migrating tasks from a hot core to a cold core [15] could end up incurring more performance loss than simply throttling hot cores, because the cold core can be more leaky due to core-to-core variations. Combined consideration of variability and temperature is necessary to address this problem [16]. There have been proposals addressing this problem reactively, based on temperature sensor measurement [15] or proactively by predicting future behavior with thermal history and taking preemptive actions [17].

Perhaps the biggest challenge for dynamic thermal management is in achieving sufficiently accurate temperature measurement. Temperature sensors used in current processors remain costly yet fairly imprecise, so the challenge is how to use a small number of on-chip temperature sensors that have limited accuracy to achieve sufficiently accurate thermal control. To make matters worse, hot spot locations change with workloads, and sensor circuits are difficult to place near hot structures, especially dense datapaths and array structures such as caches and register files. When the sensors are too far away from the actual hotspots, their accuracy falls off dramatically. These challenges are exacerbated by manufacturing variations, which can change the location and severity of hotspots. The appropriate choice of guardbands—and how rigid to make them—is also an open question, because they must protect against many failure mechanisms: timing errors, soft errors due to thermal noise, excessive leakage, and a variety of aging phenomena that develop at different temperature-dependent rates. There are a few recent studies that make promising advances on these issues [18], but more research is needed. Without sufficient accuracy, guardbands are too large, imposing high costs in performance or cooling. However, aggressive deployment of precise sensors remains costly.

B. Thermal modeling and management need to be hierarchical

Another major challenge is the multi-scale nature of heat transfer. As we have seen, temperature effects must be considered at granularities ranging from individual transistors to entire racks in a datacenter. Silicon has a spatial temperature variation as small as several microns and its temporal variations are of hundreds of microseconds to milliseconds. In comparison, packages have millimeters and seconds, servers have centimeters and minutes, and data centers have meters and hours. Designs at different levels are tightly coupled to each other, so to achieve a fully thermally optimized design, one needs to also have a tightly coupled thermal model ranging from transistors to machine rooms, and from microseconds to hours. Additionally, most reliability problems only manifest themselves over long time durations, requiring modeling and management techniques that can accommodate such long time scales without sacrificing too much precision. Brute-force modeling of fine details over long time scales is prohibitively expensive. Hierarchical thermal modeling and management is needed, requiring new ways to link models from different levels of the design hierarchy in ways that maintain accuracy at each granularity yet capture temperature evolution over long time scales. This will require collaboration from researchers

at each level. There are some initial efforts on this topic at the chip level [19], system level [3], and data center level [20].

C. Thermals are interrelated with other physical constraints

The thermal constraint is just one of several constraints that architects are facing. Manufacturing variations have already been mentioned; in addition to thermal implications, they also affect performance, power, and reliability. Power delivery limits are another major constraint. The demand for I/O and current is going up faster than the package pin count can, because the density of processing units is roughly doubling each generation (i.e. Moore's Law) while pad size cannot shrink much and the number of pads scales only slowly. Consequently, power delivery (more accurately, current delivery) to future processors becomes a real challenge. Large current and current swings also pose thermally-induced reliability issues on silicon power supply pads as well as large on-chip voltage noise. Although the emergence of on-chip voltage regulators may alleviate these power delivery constraints to some extent, system-level power constraints are present too: limits on the current that affordable batteries or power supplies can source, limits on power distribution within data centers, and so forth. So far it is still unclear which constraint will be met first, temperature or power delivery?

VI. Summary

To summarize, with technology scaling pushing the limits of affordable cooling, processor design must be temperature-aware from beginning to end, and many important research questions remain.

References

[1] P. Dadvar and K. Skadron, "Potential thermal security risks," in *Proceedings of the IEEE Semiconductor Thermal Measurement, Modeling, and Management Symposium (Semi-Therm 21)*, March 2005.

[2] K. Skadron, M. R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan, "Temperature-aware microarchitecture," in *Proc. Intl. Symp. on Computer Architecture (ISCA)*, June 2003, pp. 2–13.

[3] T. Heath, A. P. Centeno, P. George, L. Ramos, Y. Jaluria, and R. Bianchini, "Mercury and freon: Temperature emulation and management for server systems," in *Proceedings of the International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, October 2006.

[4] W. Huang, E. Humenay, K. Skadron, and M. Stan, "The need for a full-chip and package thermal model for thermally optimized IC designs," in *Proc. Intl. Symp. on Low Power Electronic Design (ISLPED)*, August 2005, pp. 245–250.

[5] W. Huang, K. Sankaranarayanan, K. Skadron, R. J. Ribando, and M. R. Stan, "Accurate, pre-rtl temperature-aware processor design using a parameterized, geometric thermal model," *IEEE Transactions on Computers*, vol. 57, no. 9, 2008.

[6] Y. Yang, Z. P. Gu, C. Zhu, R. P. Dick, and L. Shang, "Isac: Integrated space and time adaptive chip-package thermal analysis," *IEEE Transactions Computer-Aided Design*, vol. 26, no. 1, pp. 86–99, January 2007.

[7] Y. Li, D. Brooks, Z. Hu, and K. Skadron, "Performance, energy, and thermal considerations for SMT and CMP architectures," in *Proc. High Performance Computer Architecture (HPCA)*, February 2005, pp. 71–82.

[8] W. Huang, K. Skadron, S. Gurumurthi, R. Ribando, and M. R. Stan, "Exploring the thermal impact on manycore processor performance." in *Proc. of IEEE Semi-Therm Symposium*, February 2010.

[9] C. Green, A. Fedorov, and Y. Joshi, "Fluid-to-fluid spot-to-spreader (f2/s2) hybrid heat sink for integrated chip-level and hot spot-level thermal management," *ASME Journal of Electronic Packaging*, vol. 131, no. 6, June 2009.

[10] W. Huang, M. R. Stan, S. Gurumurthi, R. Ribando, and K. Skadron, "Interaction of scaling trends in processor architecture and cooling." in *Proc. of IEEE Semi-Therm Symposium*, February 2010.

[11] T. Brunschwiler, H. Rothuizen, U. Kloter, H. Reichl, B. Wunderle, H. Oppermann, and B. Michel, "Forced convective interlayer cooling potential in vertically integrated packages," in *Proceedings of the IEEE/ASME Tenth Intersociety Conference on Thermal and Thermo-mechanical Phenomena in Electronic Systems (ITHERM)*, May 2008.

[12] D. Shin, J. Kim, N. Chang, J. Choi, S.-W. Chung, and E.-Y. Chung, "Energy-optimal dynamic thermal management for green computing," in *Proceedings of IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, November 2009.

[13] "Gradient DA Inc, Case Study: http://www.gradient-da.com/resources/case-study2.php."

[14] J. Kong, S. W. Chung, and K. Skadron, "Recent thermal management techniques for microprocessors," *ACM Computing Surveys*, To appear.

[15] A. Coskun, T. S. Rosing, K. A. Whisnant, and K. C. Gross, "Static and dynamic temperature-aware scheduling for multiprocessor socs," *IEEE Transactions on VLSI*, vol. 16, no. 9, pp. 1127–1140, September 2008.

[16] E. Humenay, D. Tarjan, and K. Skadron, "Impact of process variations on multicore performance symmetry," in *Proceedings of the ACM/IEEE/EDAA/EDAC 2007 Conference on Design, Automation and Test in Europe (DATE)*, April 2007.

[17] A. K. Coskun, T. S. Rosing, and K. C. Gross, "Proactive temperature balancing for low cost thermal management in mpsocs," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, November 2008.

[18] S. Sharifi and T. S. Rosing., "Accurate direct and indirect on-chip temperature sensing for efficient dynamic thermal management," *IEEE Transactions on Computer-Aided Design*, vol. 29, no. 10, pp. 1586–1599, October 2010.

[19] Z. Hassan, N. Allec, L. Shang, R. Dick, V. Venkatraman, and R. Yang, "Multiscale thermal analysis for nanometer-scale integrated circuits," *IEEE Transactions on Computer-Aided Design*, vol. 28, no. 6, pp. 860–873, June 2009.

[20] H. F. Hamann, T. G. van Kessel, M. Iyengar, J.-Y. Chung, W. Hirt, M. A. Schappert, A. Claassen, J. M. Cook, W. Min, Y. Amemiya, V. Lopez, J. A. Lacey, and M. O'Boyle, "Uncovering energy-efficiency opportunities in data centers," *IBM Journal of Research and Development*, vol. 53, no. 3, 2009.