Software Systems Seminar,
University of Salzburg, Austria,
Summer 2007

# A Survey of Dynamic Thermal Management and Power Consumption Estimation

Andreas Naderlinger
Department of Computer Sciences
University of Salzburg, Austria
andreas.naderlinger@cs.uni-salzburg.at

## Abstract

This paper represents a survey of the research context behind the paper *Balancing Power Consumption in Multiprocessor Systems* [15] by A. Merkel and F. Bellosa, presented at EuroSys06. Power dissipation is becoming an increasing problematic side effect of power consumption in computing systems. Contemporary cooling infrastructures are close to their limits and costs are ever increasing. The survey is intended to give a brief overview of different dynamic thermal management approaches and how they are leveraged towards designing economical systems while maintaining performance. It outlines various energy estimation techniques as a basic prerequisite for efficient energy-aware management at run-time.

## 1 Introduction

Moore's law states that the number of transistors on a chip doubles about every two years or less. Unfortunately, there are no indications for the existence of a *law* that signifies the reduction of power demands or resulting power dissipation in form of heat in a similar extent. In fact, power density is becoming a major challenge in system design, as in recent years power density has doubled every three years [26]. Energy-awareness used to be in issue for mobile or embedded systems only, but as power density on chip level rises exponentially [7], problems regarding power supply and cooling arise in various areas and on different levels, ranging from chip granularity, to servers, and data centers [22].

Tremendous advances in miniaturization and clock frequencies coupled with ever increasing demands on computing power have pushed current cooling systems close to their limits. Power dissipation has become one of the most critical aspect for system design [8], as *hot spots* may lead to errors or even physical damage. Together with power dissipation also the disparity between the maximum and the typical power consumption of a processor is increasing, leading to the following dilemma: The system design must ensure not to exceed some critical temperature. However, most workloads do not exploit this limit. Figure 1 illustrates the non-linear relation between thermal dissipation and cooling costs and motivates the trend to design systems for *worst typical* application [5, 26]. Dynamic thermal management (DTM) is used to close the gap, by applying various temperature lowering techniques at runtime.

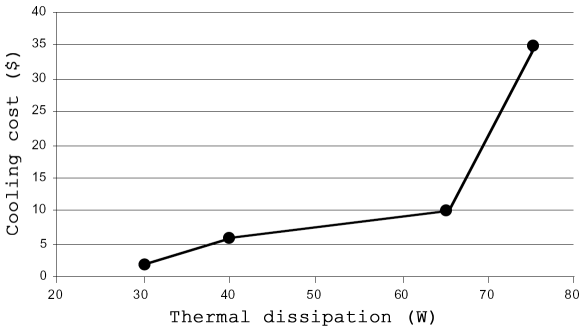


Figure 1: Cooling cost vs. thermal dissipation, according to [7]

# 2 Power awareness and optimization techniques

There are various reasons for power-awareness, e.g., effectively use limited resources for mobile devices, limiting (cooling-)costs or increasing system throughput by reducing power dissipation (and thus need for CPU throttling). Appliances are numerous as well, as this issue involves many computer system components, like HDDs, display, etc. This paper, however, is limited to processor related applications and techniques. Beside the design and architecture of hardware, software has substantial impact on power dissipation, as it drives hardware activities and thus influences energy consumption [14]. For recent processors, consumption is strongly dependent on instruction properties, such as register numbers. The paper [23] describes a simulation and profiling tool that approximates energy consumption for embedded systems and points out redesign potential for optimizing code. Shina et al. [24] managed to apply algorithmic transformations to achieve efficient energy utilization. Tan et al. [28] abstract from instruction-level and compiler techniques and propose a high-level software architecture transformation. Techniques for operating systems are also subject to numerous research in this field. For example, Zeng et al. [31] present ECOSystem, an energy-centric operating system. The main goal is extending battery lifetime, by providing a single management framework for diverse hardware resources.

# 3 Dynamic Thermal Management (DTM)

Optimizing components on both, hardware- and software side holds a great potential to reducing heat dissipation. However, these approaches must be applied from the outset - statically. Dynamic thermal management (DTM) refers to dynamic hard- and software strategies for controlling a chip's operating temperature at runtime. Beside reducing power consumption and thus temperature the aim is at keeping performance penalties as little as possible. A first step towards dynamical management was the ACPI (Advanced Configuration and Power Interface) specification [1]. ACPI puts the operating system in control of power management, instead of the BIOS, but is quite coarse-grained in its facilities. Recent DTM strategies allow for much finer-grained solutions [10, 5]. Basically, we can identify two different techniques for managing power density within a processor, temporal and spatial solutions.

## 3.1 Temporal DTM Solutions

The basic idea behind temporal solutions is to slow (throttle) or even stop computation long enough in order to allow the processor to cool down.

#### Direct feedback-driven activity reduction

Activity reduction mechanisms were proposed in various forms: voltage or frequency scaling [16], instruction cache throttling [21], or fetch-toggling (instruction fetching is stalled for several cycles) [5]. Brooks and Martonosi [5] give a comparison of several techniques. All these solutions trade in runtime performance for keeping the processors beyond a certain level of temperature. Other approaches do not affect the whole system, but operate on a more fine-grained level. [19], for example, limits the intervention to CPU-intensive tasks; thus, performance degradation does not necessarily affect interrupt processing, or tasks that do anyway not contribute to a high processor temperature, like many user-interactive applications. Huang et al. [10] try to overcome shortcomings of single, independent techniques and combine many of them in their proposed energy-management framework. The framework addresses both, energy efficiency and temperature management.

## 3.2 Spatial DTM Solutions

The focus of spatial solutions lies on minimizing performance penalties by distributing power consumption across some total system. The basic idea is that a balanced power consumption and thus a balanced heating of the total system leads to performance gains.

Many approaches are based on the following facts and observations [9, 15, 17]:

- in modern microprocessors, power dissipation is distributed unevenly, leading to localized hot spots;

- processor power (consequently temperature) describes a nonlinear relationship to input voltage;
- the relation between CPU utilization and temperature is nonlinear;
- while power consumption reacts and changes immediately, temperature in-/decreases slowly;
- if any essential resource (like register file, ALU) reaches its critical temperature, the entire core has to stop execution;
- the number of hot resources has only little impact on the cooling time.

Silicon is a relatively poor heat conductor and cannot spread heat efficiently across a die. Therefore, to avoid hot spots, heat-causing computational activities are distributed (migrated) themselves.

Such a spatial solution can be applied on very different levels of granularity. And in the ideal case, one could think of a combination of them on all levels:

### DTM within a single core

[27] and [9], for example, propose to introduce spare components of hot spot endangered resources. When the temperature of such a replicated resource reaches a certain level, computational activities should be migrated to their 'twins'. Simulations in [9] have shown that the implied overhead is by far not comparable with the resulting throughput increase. Admittedly, such replication results in heightened wiring complexity and required space and leads to under-utilized resources.

### DTM within a single chip

The current trend of on-chip parallelization by techniques like simultaneous multithreading (SMT) or chip-level multiprocessors (CMP) does everything but contributing to cooler temperatures. On the contrary, both lead to raised power density, as (1) SMT increases processor-resource utilization and (2) CMPs take the same die area for two or more cores as former superscalar CPUs with only one core. Fortunately, SMT CMPs also pose opportunities for managing power density. As the additional cores can be seen as spare resources as described above. However, the replication of a complete core allows for far more flexibility, as they can be used for executing different threads with non- or less problematic workloads in respect to power density.

Powell et al.[17] propose a technique called *heat and run* that seeks to increase the system's performance by controlling power density. Heat-and-run combines two key concepts:

**Heat-and-run thread assignment (HRTA)** HRTA prompts the operating system to assign threads to CMP cores in a way that as much different resources on the core as possible are heated up simultaneously to their critical temperature. So, the goal is to combine (co-schedule) threads with complementary resource requirements. This proceeding is based on the fact that heat transfer away from the processor is much higher than transfer among sub-components. Thus, the time required for cooling a core before overheating does not heavily depend on the number of hot core-resources. As anyway the complete core has to be stopped when one single resource runs the risk of overheating, this approach allows multiple resources to cool down at the same time. Hence, cooling time is used more effectively.

**Heat-and-run thread migration (HRTM)** Before a core has to be stopped in order to avoid overheating, HRTM is applied, which prompts the OS to migrate threads and thus heat away from the hot core. Ideally, the target core is cold or at least executes a complementary workload. While the threads keep running on a second core, the other one is cooled. HRTA balances heat generation among all functional units in each core, whereas HRTM balances heat in the whole chip.

### DTM involving multiple chips

Merkel and Bellosa apply similar techniques on a more coarse-grained level. In [15], they research into multiprocessor systems and present alternative mechanisms to throttling. Thanks to the availability of multiple processors, activity migration (on task level) becomes even more effective. Their energy-aware scheduling method is based on the Linux's load-balancing mechanism and co-

schedules tasks with different energy characteristics. Hot tasks, consuming more power, are combined with cool tasks, whereby a balanced power consumption among all CPUs in the microprocessor is achieved.

# 4 Power Estimation

All the methods mentioned above need proper information on temperate according to which expedient actions, like CPU throttling or activity migration, can be taken. Normally, CPUs are equipped with (hardware) sensors. Unfortunately, those thermal diodes provide quite low resolution, are noisy and difficult to calibrate [15, 26]. Additionally, reading the diode (e.g., via the system management bus) involves non-negligible overhead [15]. In [7], Gunther et al. give detailed insights into detection mechanisms for recent Intel processors. As more accurate sensing possibilities are often missing, several approaches try to model power dissipation as a function of the executed software (instructions) on a specific hardware platform.

For recent processor it is not possible to derive power consumption from the CPU load. The kind of instruction being executed by the processor have a crucial impact on power characteristics[11].

## 4.1 Instruction-level estimation

In 1994, Vivek Tiwari et al. [30] pioneered in the field of power estimation and proposed a technique based on instruction-level power modeling. The basic idea behind this approach is to assign a base energy cost factor to each individual processor instruction. Given a set of instructions which refer to a certain piece of software, the weighted sum represents the program's total energy consumption. However, considering only base costs does not reflect the actual power consumption, since the sequence of instruction plays an important role. Thus, the described approach also took instruction-pairs into account, as well as pipeline stalls and cache miss effects. Considering more than just two consecutive instructions (pairs) would lead to a more accurate result, but also implies a combinatorial problem. Years later, this approach was refined regarding energy model accuracy and performance [20].

## 4.2 Function-level and macro-modeling estimation

Instead of describing power consumption on instruction-level, [18] is using a *power data bank* for embedded systems that stores power information derived from simulations on function-level. As considerably parts of the code are covered by (built-in) library functions, only minor code segments have to be evaluated on a time-consuming instruction-level. Tan et al. [29] propose a power estimation technique based on macro-models. Macro-models relate power consumption to different parameters that can either be observed or derived from (high-level) programming language descriptions. While maintaining high accuracy, this approach achieves a notable performance gain compared to lower level techniques.

## 4.3 Event counters

Event (or performance) monitoring counters [2] offered by modern processors form an alternative means for estimating power consumption. These values (accessible by special registers) were originally intended for performance analysis and optimization, and reflect different processor activities. In [4], Bellosa describes the potential of performance counters in the field of power-sensitive systems. Subsequent work is described in [12, 3]. [13] describes a tool for application energy-profiling based on event counters. Isci et al. [11] also applies this technique and provides power information for more than twenty major CPU subunits.

## 4.4 Thermal models and simulation

Based on the parallels between heat transfer and electrical circuits, thermal models are used to derive temperature from power consumption. [3] describes a simple model consisting of a thermal resistor and capacitor, that is able to estimate temperature with an error of less than $1°C$ for real-world applications. *HotSpot* [26, 25], and *Wattch* [6] are approaches to model thermal behavior in power- and performance simulators on architectural level. As they are parameterizable, they can be easily adapted for different microarchitectures. *SoftWatt* [8] is an alternative approach used for complete machine simulation.

# 5 Conclusion

Energy-awareness is not any longer solely an issue for mobile devices with limited resources. The exponential rise of cooling costs stemming from ever increasing demand on computing power and clock frequencies have necessitated a rethink of traditional worst-case cooling infrastructures. Typical-case architectures together with methods arranging for peak-load (worst-case) scenarios are becoming more and more accepted. This paper provides a brief overview of some dynamic thermal management techniques. Special focus is on spatial DTM approaches maintaining performance, and on power consumption estimation techniques.

# References


[1] Advanced Configuration and Power Interface Specification, http://www.acpi.info.

[2] Jennifer M. Anderson, Lance M. Berc, Jeffrey Dean, Sanjay Ghemawat, Monika R. Henzinger, Shun-Tak A. Leung, Richard L. Sites, Mark T. Vandevoorde, Carl A. Waldspurger, and William E. Weihl. Continuous profiling: where have all the cycles gone? *ACM Trans. Comput. Syst.*, 15(4):357–390, 1997.

[3] F. Bellosa, A. Weissel, M. Waitz, and S.Kellner. Event driven energy accounting for dynamic thermal management. In *COLP '03: Proceedings of the workshop on Compilers and Operating Systems for Low Power*, 2003.

[4] Frank Bellosa. The benefits of event: driven energy accounting in power-sensitive systems. In *EW 9: Proceedings of the 9th workshop on ACM SIGOPS European workshop*, pages 37–42, New York, NY, USA, 2000. ACM Press.

[5] David Brooks and Margaret Martonosi. Dynamic thermal management for high-performance microprocessors. In *HPCA '01: Proceedings of the 7th International Symposium on High-Performance Computer Architecture*, page 171, Washington, DC, USA, 2001. IEEE Computer Society.

[6] David Brooks, Vivek Tiwari, and Margaret Martonosi. Wattch: A framework for architectural-level power analysis and optimizations. In *Proceedings of the 27th Annual International Symposium on Computer Architecture*, 2000.

[7] Stephen H. Gunther, Frank Binns, Douglas M. Carmean, and Jonathan C. Hall. Managing the impact of increasing microprocessor power consumption. *Intel Technology Journal*, 2001.

[8] Sudhanva Gurumurthi, Anand Sivasubramaniam, Mary Jane Irwin, Narayanan Vijaykrishnan, Mahmut T. Kandemir, Tao Li, and Lizy Kurian John. Using complete machine simulation for software power estimation: The softwatt approach. In *HPCA*, pages 141–150, 2002.

[9] Seongmoo Heo, Kenneth Barr, and Krste Asanović. Reducing power density through activity migration. In *ISLPED '03: Proceedings of the 2003 international symposium on Low power electronics and design*, pages 217–222, New York, NY, USA, 2003. ACM Press.

[10] M. Huang, J. Renau, S-M. Yoo, and Josep Torrellas. A Framework for Dynamic Energy Efficiency and Temperature Management. In *33rd International Symposium on Microarchitecture*, December 2000.

[11] Canturk Isci and Margaret Martonosi. Runtime power monitoring in high-end processors: Methodology and empirical data. In *MICRO 36: Proceedings of the 36th annual IEEE/ACM International Symposium on Microarchitecture*, page 93, Washington, DC, USA, 2003. IEEE Computer Society.

[12] Russ Joseph and M. Martonosi. Run-time power estimation in high-performance microprocessors. In *The International Symposium on Low Power Electronics and Design ISLPED'01*, August 2001.

[13] I. Kadayif, T. Chinoda, M. Kandemir, N. Vijaykirsnan, M. J. Irwin, and A. Sivasubramaniam. vec: virtual energy counters. In *PASTE '01: Proceedings of the 2001 ACM SIGPLAN-SIGSOFT workshop on Program analysis for software tools and engineering*, pages 28–31, New York, NY, USA, 2001. ACM Press.

[14] Tao Li and Lizy Kurian John. Run-time modeling and estimation of operating system power consumption. In *Proceedings of the International Conference on Measurement and Modeling of Computer Systems SIGMETRICS'2003*, June 2003.

[15] Andreas Merkel and Frank Bellosa. Balancing power consumption in multiprocessor systems. In *First ACM SIGOPS EuroSys Conference*, Leuven, Belgium, April 18–21 2006.

[16] T. Pering and R. Broderson. The simulation and evaluation of dynamic voltage scaling algorithms. In *Proceedings of the International Symposium on Low-Power Electronics and Design ISLPED'98*, June 1998.

[17] 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.

[18] Gang Quy, Naoyuki Kawabez, Kimiyoshi Usamiz, and Miodrag Potkonjaky. Function-level power estimation methodology for microprocessors. In *Design Automation Conference*, 2000.

[19] E. Rohou and M. Smith. Dynamically managing processor temperature and power. In *2nd Workshop on FeedbackDirected Optimization*, Nov 1999.

[20] A. Sama, M. Balakrishnan, and J. F. M. Theeuwen. Speeding up power estimation of embedded software. In *Proc. Int. Symp. Low Power Electronics and Design*, pages 191–196, 2000.

[21] Hector Sanchez, Belli Kuttanna, Tim Olson, Mike Alexander, Gian Gerosa, Ross Philip, and Jose Alvarez. Thermal management system for high performance powerpctm microprocessors. In *COMPCON '97: Proceedings of the 42nd IEEE International Computer Conference*, page 325, Washington, DC, USA, 1997. IEEE Computer Society.

[22] Ratnesh K. Sharma, Cullen E. Bash, Chandrakant D. Patel, Richard J. Friedrich, and Jeffrey S. Chase. Balance of power: Dynamic thermal management for internet data centers. *IEEE Internet Computing*, 9(1):42–49, 2005.

[23] T. Simunic, L. Benini, and G. De Micheli. Energy-efficient design of battery-powered embedded systems. In *Proceedings of the International Symposium on Low-Power Electronics and Design ISLPED'98*, June 1998.

[24] Amit Sinha, Alice Wang, and Anantha P. Chandrakasan. Algorithmic transforms for efficient energy scalable computation. In *ISLPED '00: Proceedings of the 2000 international symposium on Low power electronics and design*, pages 31–36, New York, NY, USA, 2000. ACM Press.

[25] K. Skadron, M. Stan, M. Barcella, A. Dwarka, W. Huang, Y. Li, Y. Ma, A. Naidu, D. Parikh, P. Re, S. Velusamy, H. Zhang, and Y. Zhang. Hotspot: Techniques for modeling thermal effects at the processorarchitecture level. In *Proceedings of the 8th International Workshop on Thermal Investigations of ICs and Systems (THERMINICS-8)*, 2002.

[26] Kevin Skadron, Mircea R. Stan, Wei Huang, Sivakumar Velusamy, Karthik Sankaranarayanan, and David Tarjan. Temperature-aware computer systems: Opportunities and challenges. *IEEE Micro*, 23(6):52–61, 2003.

[27] Kevin Skadron, Mircea R. Stan, Wei Huang, Sivakumar Velusamy, Karthik Sankaranarayanan, and David Tarjan. Temperature-aware microarchitecture. In *Proceedings of the 30th International Symposium on Computer Architecture (ISCA'03)*, June 2003.

[28] T. K. Tan, A. Raghunathan, and N. K. Jha. Software architectural transformations: A new approach to low energy embedded software. In *DATE '03: Proceedings of the conference on Design, Automation and Test in Europe*, page 11046, Washington, DC, USA, 2003. IEEE Computer Society.

[29] T. K. Tan, Anand Raghunathan, Ganesh Lakshminarayana, and Niraj K. Jha. High-level

software energy macro-modeling. In *Design Automation Conference*, pages 605–610, 2001.

[30] V. Tiwari, S. Malik, and A. Wolfe. Power analysis of embedded software: a first step towards software power minimization. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2(4):437–445, 1994.

[31] Heng Zeng, Xiaobo Fan, Carla Ellis, Alvin Lebeck, and Amin Vahdat. ECOSystem: Managing energy as a first class operating system resource. In *Tenth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS X)*, October 2002.