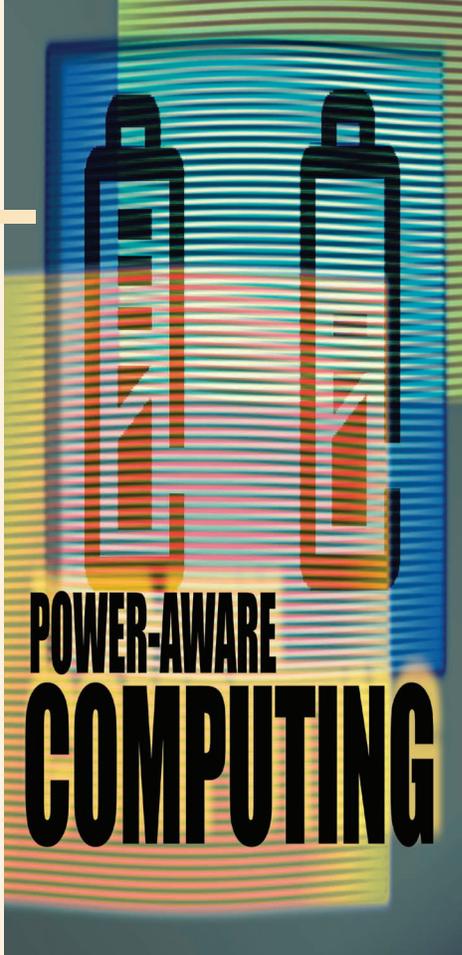


Power-Aware Computing

Performance, complexity, cost, and power tradeoffs have created exciting challenges and opportunities in the rapidly changing field of power-aware computing.



**POWER-AWARE
COMPUTING**

Mircea R. Stan

Kevin Skadron
University of Virginia

In the past 50-some years—the entire lifetime of the electronic computer—the mantra computer designers and users have chanted in unison has been “faster... smaller... cheaper...,” with the more recently added “and lower power...” significantly complicating the whole picture. The tradeoffs among performance, complexity, cost, and power have created exciting challenges and opportunities—not to mention long, sleepless nights—for everyone involved in this rapidly changing field.

In high-performance systems, power-aware design techniques aim to maximize performance under power dissipation and power consumption constraints—the system’s power envelope. At the other extreme, low-power design techniques try to reduce power or energy consumption in portable equipment to meet a desired performance or throughput target. All the power a system consumes eventually dissipates and transforms into heat. The power dissipation and related thermal issues affect performance, packaging, reliability, environmental-impact, and heat removal costs; the power and energy consumption affect power delivery costs, performance, and reliability, and they relate directly to size and battery life for portable devices.

POWER TRADEOFFS

Most computing systems have at least two modes of operation: an active mode, when useful computation takes place, and an idle mode, when the system is inactive. It is acceptable to have higher power con-

sumption in active mode as a tradeoff to increased performance, but any power consumed when the system is idle is a complete waste and ideally should be avoided by turning the system off.

A typical current system is so complex that parts of it will likely be inactive even during active periods, and they can be turned off to reduce power with no impact on performance. The introduction of finer-grained power modes—for example, run at half speed and with lower power supply voltage—can further refine such simple strategies to reduce power in time (turn off the system during idle times) and space (turn off inactive elements), leading to more complex tradeoffs in terms of performance and verification costs.

Although such power-aware strategies can be activated either in hardware or software, they usually are activated in software with hardware support. The techniques also can be static (compile-time) or online (runtime), with online methods being more flexible but generally having worse results than those achieved with a profile-guided static method.

How to compare such different power-aware computing methods against each other poses an important question. Depending on the application—peak power, dynamic power, average power, energy, energy-delay product, energy-delay-squared product, power density—we can use several figures of merit for this purpose.

The two metrics that have proved most useful until now are the energy-delay product (inversely

The energy-delay-squared product provides a convenient first-order approximation for CMOS circuits.

proportional to MIPS²/watt) and the energy-delay-squared product (inversely proportional to MIPS³/watt). Independent of the power supply voltage, the energy-delay-squared product is useful for architecture-level optimizations. This provides a convenient first-order approximation for CMOS circuits, for which dynamic energy is proportional to V_{dd}² and delay is inversely proportional to V_{dd}, hence they cancel out.

Once the architecture work has been completed, the energy-delay product is a convenient metric for choosing an optimal voltage at the circuit level that will provide the highest performance level for a given power envelope or the lowest power for a given throughput or delay.

TEMPERATURE-AWARE COMPUTING

Recently, researchers have recognized that thermal issues merit a thorough investigation as more than just an extension of power-aware methods. Fundamentally, since temperature is a by-product of power dissipation, power-aware design and the emerging field of temperature-aware design¹ are intrinsically related, but they also have significant differences.

First, temperature is proportional to power density, not just power, so methods to reduce thermal effects can either reduce power, increase area, or both. The common use of heat spreaders in modern high-performance microprocessors provides one example of increasing area to reduce power density and deal with thermal effects.

Second, even if power density determines temperature, instantaneous power density by itself cannot serve as a proxy for temperature because the long time constants in the thermal domain tend to filter out any fast changes. Even if there is filtering, average power density cannot be used as a proxy for temperature either because there are significant thermal gradients in space and time that cannot be inferred without modeling temperature and heat transfer directly.

The processor architecture and system architecture domains are unique in their ability to use runtime knowledge of application behavior and the chip's thermal status to control execution rate, distribute the workload, and extract instruction-level parallelism. On-chip temperature sensors can provide information about local hot spots and spatial or temporal temperature gradients. The architecture can combine this information with dynamic information about ILP and workload characteristics to precisely regulate temperature while minimizing performance loss.²

Thus far, research on temperature-aware architecture has focused on *dynamic thermal management*. DTM recognizes that if the thermal package is designed for worst-case power dissipation, it must be designed for the most severe hot spot that could potentially arise. Yet these worst-case scenarios are rare and lead to overly engineered solutions. Instead, researchers can use a less expensive package designed for the worst “typical” or “interesting” workload. Then, an autonomous runtime response in the chip itself must handle the thermal stress, guaranteeing to reduce power densities far and fast enough to maintain temperature regulation. The Intel Pentium-4 follows this approach, providing a thermal package design for 20 percent less than absolute worst case.³

There is a clear need for a robust thermal modeling infrastructure to explore temperature-aware computing solutions at the architecture level. Our recently introduced HotSpot thermal model (<http://lava.cs.virginia.edu/HotSpot>) provides an accurate means of modeling temperature with enough granularity to observe thermal gradients in space and time. HotSpot directly interfaces with the most popular tools the architecture community uses, such as the SimpleScalar performance modeling tool and the Wattch power estimation tool.

Understanding and exploiting thermal effects is critical because of their impact on packaging, reliability issues, performance, and leakage power. Electromigration, aging of the thin oxide, mechanical failure due to thermal gradients, and expansion coefficients are the main factors that result in increased failure rates under thermal stress. Performance is lower and leakage power exponentially higher at high temperatures, which suggests that effectively dealing with thermal issues can increase performance and reduce power consumption simultaneously. Because of this, researchers are investigating active cooling techniques to control temperature, which may become a requirement for future processes.

IN THIS ISSUE

Several key elements form a computing system: processors; memory; peripheral devices; input, output, and communication devices; power delivery; and conditioning. The power tradeoffs for each of these components are quite specific and typically considered separately. The articles selected for inclusion in this special issue illustrate some of these points of view.

In “Energy Management for Commercial Servers,” Charles Lefurgy and colleagues look at

the complex situation encountered with some of the most complicated high-performance computing systems, for which, historically, power consumption has not been a strong constraint. However, sharp increases in power consumption and the difficulties encountered in simply trying to deliver that power and then remove the resulting heat mean that nowadays even these high-performance servers must make a concerted effort at all levels of abstraction, in both hardware and software, to keep power under control. This article describes techniques that combine hardware and software power management to smartly deliver heterogeneous loads to the different processors and to reduce the intrinsic power that processors and memory devices consume.

“Dynamically Tuning Processor Resources with Adaptive Processing” by David H. Albonesi and colleagues provides an excellent complementary view by focusing on the processor itself. Depending on the application, current microprocessors are so complex that their resources usually cannot be fully utilized. The authors propose that turning off parts of some critical elements—an associative cache, part of the issue queue, or the register file—will result in a power reduction with little performance penalty. The authors show very promising results with this approach, which they refer to as “adapting the complexity.”

As an example of the issues affecting a larger class of peripheral devices, “Reducing Disk Power Consumption in Servers with DRPM” by Sudhanva Gurumurthi and coauthors tackles the complex puzzle of power-aware design at the system level. To reduce the significant penalty paid when a disk drive goes between idle and full-speed mode—especially since server disk drives rarely can be totally idle—the authors propose the *dynamic rotations per minute* scheme, an essentially fine-grained power mode scheme for disk drives. Since power consumption is directly proportional to rotation speed, DRPM leads to energy savings because the disks always rotate at a speed suitable to the current application.

“Leakage Current: Moore’s Law Meets Static Power” by Nam Sung Kim and colleagues considers the important issues associated with the relationship between dynamic and static power. When the circuit is active, it consumes both dynamic and static power, but when it is idle, it consumes only static power. From the idle-mode perspective then, static power should be as small as possible; in active mode, however, the tradeoffs are such that the high-performance is obtained when there is also sig-

nificant static power. Power-aware methods then need to address the contradicting requirements between active and idle modes. The authors discuss fine-grained power modes such as snooze, the significant problem of cache memory leakage—which is dominant for such circuits, and software and technology solutions to some of these issues.

Finally, in “Battery Modeling for Energy-Aware System Design,” Ravishankar Rao and coauthors consider power delivery issues for portable devices. The ability to model batteries with different chemistry and discharge characteristics is essential to optimizing the battery lifetime for a portable device, sometimes with surprising results. It is widely believed that a minimum energy solution is optimal for increasing battery lifetime. However, a more accurate model shows this is not always so. For example, a scenario in which the battery has distinct periods of inactivity during which it can “recover” its charge can lead to a longer lifetime than an equivalent constant discharge current case, even if the total energy is slightly higher. This article also explores many other scenarios, providing information useful to those working in the area of energy-aware design for portable systems.

In addition to these five articles, this special issue includes two authored sidebars describing adaptive processing techniques: “Managing Multiple Low-Power Adaptation Techniques: The Positional Approach” by Michael C. Huang and coauthors and “GRACE: A Cross-Layer Adaptation Framework for Saving Energy” by Daniel Grobe Sachs and colleagues. “Energy Conservation in Clustered Servers,” an authored sidebar by Ricardo Bianchini and Ram Rajamony, discusses strategies for managing energy in Web-server clusters.

We would have liked to include papers on input/output devices such as LCD and temperature-aware computing in this special issue but, unfortunately, page limits forced us to defer these topics and others to a future issue.

For more information on power-aware computing, the main conference is the annual “International Symposium on Low Power Electronics and Design” (http://portal.acm.org/browse_dl.cfm?linked=1&part=series&idx=SERIES111&coll=portal&dl=ACM). In addition, all major conferences have special sessions on this topic, and most journals and magazines have published special issues dedicated to low power.

Modeling batteries with different chemistry and discharge characteristics optimizes the battery lifetime for a portable device.

Due to page limits, we could only accept for inclusion in this special issue five papers from among the rich set of 25 submissions. This means that some excellent papers were not selected for publication. We thank all the authors who submitted their excellent work, whether their paper was included in this issue or not. ■

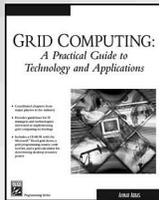
Mircea R. Stan is an associate professor of electrical and computer engineering at the University of Virginia. His research interests include low-power VLSI, temperature-aware computing, mixed-mode analog and digital circuits, computer arithmetic, embedded systems, and nanoelectronics. Stan received a PhD in electrical and computer engineering from the University of Massachusetts at Amherst. He is a senior member of the IEEE Computer Society, the IEEE, the ACM, and Usenix. Contact him at mrs8n@ee.virginia.edu.

Kevin Skadron is an assistant professor in the Department of Computer Science at the University of Virginia. His research interests include power and thermal issues, branch prediction, and techniques for fast and accurate microprocessor simulation. Skadron received a PhD in computer science from Princeton University. He is a member of the IEEE Computer Society, the IEEE, and the ACM. Contact him at skadron@cs.virginia.edu.

References

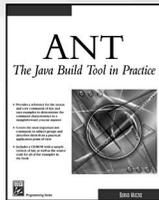
1. K. Skadron et al., "Temperature-Aware Microarchitecture," *Proc. Int'l Symp. Computer Architecture (ISCA 2003)*, IEEE CS Press, 2003, pp. 2-13.
2. J. Srinivasan and S.V. Adve, "Predictive Dynamic Thermal Management for Multimedia Applications," *Proc. 17th Ann. ACM Int'l Conf. Supercomputing (ICS03)*, ACM Press, 2003.
3. S.H. Gunther et al., "Managing the Impact of Increasing Microprocessor Power Consumption," *Intel Technology J.*; www.intel.com/technology/itj/q12001/articles/art_4.htm.

COMPUTER BOOKS for COMPUTING SUCCESS



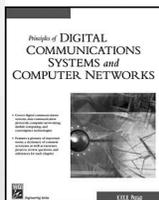
1-58450-276-2 \$49.95

A pragmatic overview of all aspects of grid computing technology.



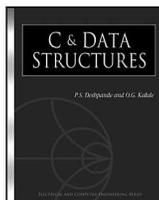
1-58450-248-7 \$39.95

A desktop guide and reference for this Java-based application building tool.



1-58450-329-7 \$59.95

A concise digital communications and networking reference for telecom professionals.



1-58450-338-6 \$59.95

A complete guide and reference to the C language and how to implement it effectively in simple and complex data structures.

Available at Amazon, Borders, Barnes & Noble, and other fine retailers



(800) 382-8505

www.charlesriver.com

20%
OFF Web
Orders

Get access

to individual IEEE Computer Society documents online.

More than 67,000 articles and conference papers available.

US\$9 per article

for members

US\$19 for nonmembers



<http://computer.org/publications/dlib>

IEEE
COMPUTER
SOCIETY