DOI:10.1145/1721654.1721673

Prior work on power management reflects recurring themes that can be leveraged to make future systems more energy efficient.

BY PARTHASARATHY RANGANATHAN

Recipe for Efficiency: Principles of Power-Aware Computing

POWER AND ENERGY are key design considerations across a spectrum of computing solutions, from supercomputers and data centers to handheld phones and other mobile computers. A large body of work focuses on managing power and improving energy efficiency. While prior work is easily summarized in two words—"Avoid waste!"—the challenge is figuring out where and why waste happens and determining how to avoid it. In this article, I discuss how, at a general level, many inefficiencies, or waste, stem from the inherent way system architects address the complex trade-offs in the system-design process. I discuss common design practices that lead to power inefficiencies in typical systems and provide an intuitive categorization of high-level approaches to addressing them. The goal is to provide practitioners—whether in systems, packaging, algorithms, user interfaces, or databases—a set of tools, or "recipes," to systematically reason about and optimize power in their respective domains.

If you are a user of any kind of computing device, chances are you can share a personal anecdote about the importance of power management in helping control the electricity (energy) it consumes. On mobile devices, this translates directly into how long the battery lasts under typical usage. The battery is often the largest and heaviest component of the system, so improved battery life also enables smaller and lighter devices. Additionally, with the increasing convergence of functionality on a single mobile device (such as phone + mp3 player + camera + Web browser), battery life is a key constraint on its utility. Indeed, longer battery life is often the highest-ranked metric in user studies of requirements for future mobile devices, trumping even increased functionality and richer applications.

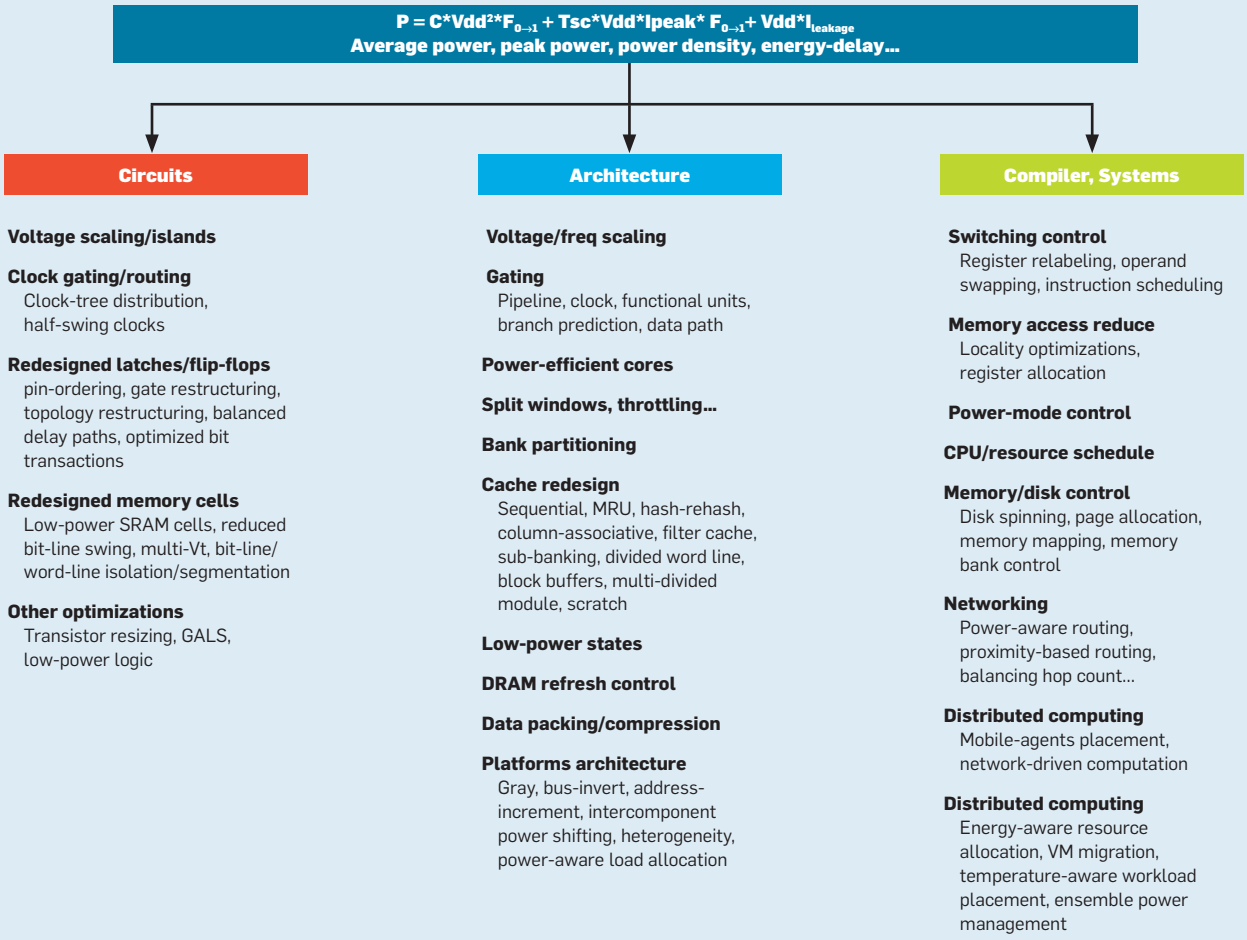
Power management is also important for tethered devices (connected to a power supply). The electricity consumption of computing equipment in a typical U.S. household runs to several hundred dollars per year. This cost is vastly multiplied in business enterprises. For example, servers in Google's data centers have been estimated to consume millions of dollars

» key insights

- The energy efficiency of today's systems can be improved by at least an order of magnitude.
- A holistic look at how systems use power and heat reveals new "recipes" to help optimize consumption and avoid wasting precious resources for a given task.
- Future power management will include nontraditional approaches, including crossing individual layers of design and spending more power to save power.

ILLUSTRATION BY JEAN-FRANCOIS PODEVIN

Overview of previous work on power management.



in electricity costs per year.[10] IT analysis firm IDC (http://www.idc.com/) estimates the total worldwide spending on power management for enterprises was likely a staggering $40 billion in 2009. Increased power consumption can also lead to increased complexity in the design of power supplies (and power distribution and backup units in larger systems) that also add costs.

Another challenge associated with power consumption in systems is the waste heat they generate; consequently, the term "power management" also includes the heat management in systems. Such heat is often a greater problem than the amount of electricity being consumed. To prevent the heat from affecting the user or the system's electronics, systems require increasingly complex thermal packaging and heat-extraction solutions, adding more costs. For large systems like supercomputers and data centers, such costs often mean an additional dollar spent on cooling for every dollar spent on electricity. This effect is captured in a metric called "power usage effectiveness," or PUE,[13] developed by the Green Grid, a global consortium of IT companies seeking to improve energy efficiency in data centers. Heat dissipation in systems also has implications for the compaction and density of computing systems, as in blade-server configurations.

Studies, most notably concerning servers and hard-disk failures, have shown that operating electronics at temperatures that exceed their operational range can lead to significant degradation of reliability; for example, the Uptime Institute, an industry organization that tracks data-center trends (http://www.uptimeinstitute.org/), has identified a 50% increased chance of server failure for each 10°C increase over 20°C[15]; similar statistics have also been shown over hard-disk lifetimes.[1,4]

Finally, power management in computing systems has environmental implications. Computing equipment in the U.S. alone is estimated to consume more than 20 million gigajoules of energy per year, the equivalent of four-million tons of carbon-dioxide emissions into the atmosphere.[10] Federal agencies have identified energy-consumption implications for air quality, national security, climate change, and electricity-grid reliability, motivating several initiatives worldwide from governmental agencies, including the Environmental Protection Agency in the U.S. (http://www.epa.gov/), Intelligent Energy Europe (ec.europa.eu/energy/intelligent/), Market Transformation Program in the U.K. (http://efficient-products.defra.gov.uk/cms/market-transformation-programme/), and Top Runner (http://www.eccj.or.jp/top_runner/index.html) in Japan, and from industry consortiums, including SPEC (http://www.spec.org/), Green-

Grid (http://www.thegreengrid.org/), and TPC (http://www.tpc.org/tpc_energy/default.asp) on improving energy efficiency, or minimizing the amount of energy consumed for a given task.

The importance of power management is only likely to increase in the future. On mobile devices, there is a widening gap between advances in battery capacity and anticipated increases in mobile-device functionality. New battery technologies (such as fuel cells) might address it, but designing more power-efficient systems will still be important. Energy-review data from the U.S. Department of Energy (http://www.eia.doe.gov/) points to steadily increasing costs for electricity. Indeed, for data centers, several reports indicate that costs associated with power and cooling could easily overtake hardware costs.[2,14] Increased compaction (such as in future predicted blade servers) will increase power densities by an order of magnitude within the next decade, and the increased densities will start hitting the physical limits of practical air-cooled solutions. Research is ongoing in alternate cooling technologies (such as efficient liquid cooling), but it will still be important to be efficient about generating heat in the first place. All of this requires better power management.

The goal is to provide practitioners a set of tools, or "recipes," to systematically reason about and optimize power in their respective domains.

How to Respond

Much prior work looked at power management and energy efficiency; the figure here outlines key illustrative solutions in the literature across different levels of the solution stack in process technology and circuits, architecture and platforms, and applications and systems design. A detailed discussion of the specific optimizations is not my intent here, and, indeed, several tutorial articles[6,10,11] and conferences that focus solely on power, including the International Symposium on Low Power Electronics and Design (http://www.islped.org/) and the Workshop on Power Aware Computing and Systems (aka HotPower; http://www.sigops.org/sosp/sosp09/hotpower.html), provide good overviews of the state of the art in power management. This rich body of work examining power management and energy efficiency can be broadly categorized across different levels of the solution stack (such as hardware and software), stages of the life cycle (such as design and runtime), components of the system (such as CPU, cache, memory, display, interconnect, peripherals, and distributed systems), target domains (such as mobile devices, wireless networks, and high-end servers), and metrics (such as battery life and worst-case power). Much prior work concerns electrical and computer systems engineering, with a relatively smaller amount in the core areas of computer science. The prior focus on power and energy challenges at the hardware and systems levels is natural and central, but, in the future, significant improvements in power and energy efficiency are likely to result from also rethinking algorithms and applications at higher levels of the solution stack. Indeed, discussions in the past few years on the future of power management focused this way.[9,12]

In spite of the seemingly rich diversity of prior work on power management, at a high level, the common theme across all solutions is "Avoid wasted energy!" Where the solutions differ is in the identification and intuition needed for specific sources of inefficiency, along with the specific mechanisms and policies needed to target these inefficiencies. This observation raises interesting questions: What general recurring high-level trends lead to these inefficiencies at different levels of the system? And what common recurring high-level approaches are customized in the context of specific scenarios? The ability to answer supports the beginnings of a structure to think about power management in a more systematic manner and potentially identify opportunities for energy efficiency beyond traditional platform-centric domains.

Sources of Waste

It is easy to imagine that there is a certain minimum amount of electrical energy needed to perform a certain task and a corresponding minimum amount of heat that must be extracted to avoid thermal problems. For example, R.N. Mayo et al.[8] performed simple experiments to measure the energy consumption of common mobile tasks (such as listening to music, making a phone call, sending email

and text messages, and browsing the Web) implemented on different devices (such as cellphones, MP3 players, laptops, and PCs) and observed two notable results: There is a significant difference in energy efficiency, often 10- to a hundredfold, across different systems performing the same task. And there are variations in the user experience across devices, but even when focused on duplicating the functionality of the best-performing system, these experiments showed it was impossible to do so at the same energy level on a different worse-performing system.

Why do some designs introduce additional inefficiencies over and above the actual energy required for a given task? My observation is that these inefficiencies are often introduced when the system design must reconcile complex trade-offs that are difficult to avoid. For example, systems are often designed for the most general case, most aggressive workload performance, and worst-case risk tolerance. Such designs can lead to resource overprovisioning to better handle transient peaks and offer redundancy in the case of failure. Moreover, individual components of a broader system are often designed by different teams (even by different vendors) without consideration for their use with one another. Individual functions of a system are also designed modularly, often without factoring their interactions with one another, adding further inefficiencies. Further, traditional designs focus primarily on system performance. This approach has sometimes led to resource-wasteful designs to extract small improvements in performance; with today's emphasis on energy costs, these small improvements are often overshadowed by the costs of power and heat extraction. Similarly, additional inefficiencies are introduced when the system design takes a narrow view of performance (vs. actual end-user requirements) or fails to address total cost of ownership, including design and operational costs.

General-purpose solutions. General-purpose systems often provide a better consumer experience; for example, most users prefer to carry a single converged mobile device rather than several separate devices (such as phone, camera, and MP3 player or GPS unit). Additionally, the exigencies of volume economics further motivate vendors to develop general-purpose systems; a product that sells in the millions of units is usually cheaper to make than, say, a product that sells in the hundreds of units.

An insidious problem is when each layer of the stack makes worst-case assumptions about other layers in the stack, leading to compound inefficiencies.

By definition, general-purpose systems must be designed to provide good performance for a multitude of different applications. This requirement results in designers using the "union" of maximum requirements of all application classes. For example, a laptop that targets a DVD-playback application might incorporate a high-resolution display and powerful graphics processor. When the same laptop is used for another task (such as reading email), the high-power characteristics of the display and graphics processor might not be needed. However, when the laptop is designed for both workloads, most designs typically include a display with the characteristics of the most aggressive application use, in this case, a high-resolution display that plays DVD movies well. Lacking adequate design thought into how energy consumption might be adapted to different kinds of tasks, such an approach often leads to significant power inefficiencies. Another example is in the data center, where optimizing for both mission-critical and non-mission-critical servers in the same facility can lead to significant inefficiencies in terms of cooling costs. Similar conflicting optimizations occur when legacy solutions must be supported on newer systems.

Planning for peaks and growth. Most workloads go through different phases when they require different performance levels from the system. For example, several studies have reported that the average server utilization in live data centers can be low (often 10%–30%). Mobile systems have also been found to spend a significant fraction of their time in idle mode or using only a small fraction of their resources.

However, most benchmarks (the basis of system design) are typically structured to stress worst-case performance workloads irrespective of how the system is likely to be used in prac-

tice. Consequently, many systems are optimized for the peak-performance scenario. In the absence of designs that proportionally scale their energy with resource utilization, the result can be significant inefficiencies. For example, many power supplies are optimized for peak conversion efficiency at high loads. When these systems are operated at low loads, the efficiency of conversion can drop dramatically, leading to power inefficiencies.

Similar overprovisioning occurs when planning for the future. Most computing systems are designed for three-to-five-year depreciation cycles, and in the case of larger installations, like data centers, even longer. Systems must be designed to ensure that sufficient capacity is built in to meet incremental growth needs. On many systems, overprovisioning also leads to inefficiencies when the system is not operating at the resource-utilization capacities that account for future growth. For example, a data center with cooling provisioned for one megawatt of operational power, but operating at only 100 kilowatts of power consumption, is significantly more inefficient than a data center with cooling provisioned for, say, 150 kilowatts of operational power and operating at 100 kilowatts of actual power consumption.

Design process structure. Current system-design approaches generally follow a structured process. System functionality is divided across multiple hardware components (CPU, chipset, memory, networking, and disk in a single system or different individual systems in a cluster) and software components (firmware, virtualization layer, operating systems, and applications). Even within a component (such as the networking stack), there are often multiple layers with well-defined abstractions and interfaces. Power management is usually implemented within these well-defined layers but often without consideration for the interaction across the layers. However, such modular designs or local optimizations might be suboptimal for global efficiency without communication across layers. An insidious problem is when each layer of the stack makes worst-case assumptions about other layers in the stack, leading to compound inefficiencies.

Information exchange across layers often enables better power optimization. For example, a power-management optimization at the physical layer of a wireless communication protocol that is aware of higher-level application activity can be more efficient than one that is oblivious to higher-level application activity. Similarly, a power-management solution that optimizes at an ensemble level (such as across different components in a system or different systems in a cluster) can be more efficient.

Similar problems exist at other boundaries of the system architecture. For example, the power management of servers is handled by the IT department, while the cooling infrastructure is often handled by a separate facilities department. This organizational structure can lead to inefficiencies as well. For example, a cooling solution that is aware of the nonuniformities in power consumption (and consequent heat generation) can be more efficient than a solution that is not.

Inefficiencies that result from layering can also be found at other places in the overall solution architecture. For example, in a classic client-server architecture, selectively exchanging information between the clients and servers has been shown to be beneficial for energy optimizations at both levels.

Tethered-system hangover. In this final design practice that leads to inefficiencies, the inefficiencies are mainly a reflection of the relentless drive to achieve higher performance, often following the assumption that there is no constraint on power, particularly by tethered systems (plugged into a power supply when in use) with no immediate consideration of battery life. For example, historically, many processor-architecture designs have included optimizations that achieved incremental performance improvements inconsistent with the amount of additional power consumed to implement the solutions. Similar tradeoffs are seen in designs for high availability at the expense of energy (such as triple modular redundancy running three concurrent executions of the same task to ensure no possibility of downtime).

Additional examples include designs with user interfaces that identify the content of interest to the user; expending energy in these areas can be more energy efficient than designs that focus on metrics like refresh rate.

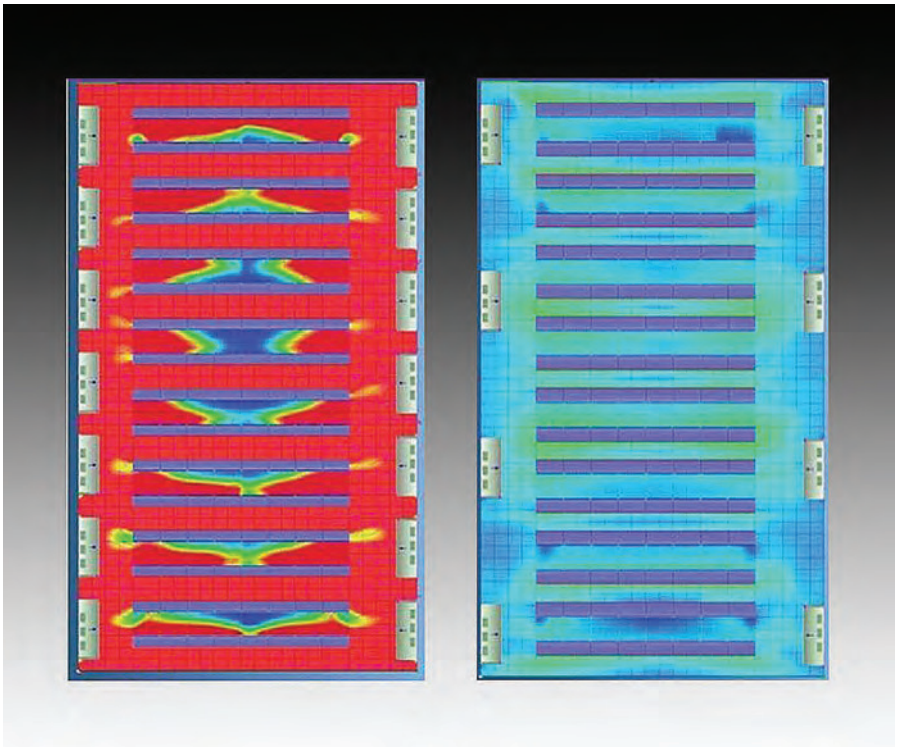


IMAGE COURTESY OF INTERNATIONAL BUSINESS MACHINES CORPORATION

IBM uses thermal analysis to test and create "green" configurations of its iDataPlex system to deliver optimal energy efficiency, as shown on the right.

Similarly, designs that focus on energy delay may be significantly more energy efficient but with only a marginal difference in performance from pure performance-centric designs. In general, several significant power inefficiencies in today's systems stem from a design focus that does not sufficiently address total cost of ownership and ultimate end-user experience, but rather focuses disproportionately on one or more narrow metrics.

How to Reduce Waste

Once these inefficiencies are identified, the next step is to identify approaches to reduce them that fall into 10 broad categories:

Use a more power-efficient alternative. These approaches include replacing a system component with a more power-efficient alternative that performs the same task with less energy. For example, more energy-efficient nonvolatile memory can replace a disk drive, and optics can replace conventional networking. A more power-efficient alternative might sometimes involve adding the right hooks to enable the approaches discussed later. For example, replacing a display with a single backlight with an alternate display that provides more fine-grain control of power can, in turn, enable power optimizations that turn off unused portions of the display. Choosing a power-efficient alternative often involves other trade-offs, possibly due to costs or performance; otherwise, the design would have used the power-efficient option in the first place.

Create "energy proportionality" by scaling down energy for unused resources. These approaches involve turning off or dialing-down unused resources proportional to system usage, often called "energy proportionality"[2] or "energy scale-down."[8] Automatically turning off unused resources requires algorithms that respond to the consequences of turning off or turning down a system (such as by understanding how long it takes to bring the system back on again). If a single component or system lacks the option to be scaled-down, the optimization is sometimes applied at the ensemble level; examples of ensemble-level scale-down include changing traffic routing to turn off unused switches and virtual-machine consolidation to coalesce workloads into a smaller subset of systems in a data center.

Decades ago, Nobel physicist Richard Feynman implied we should be able to achieve the computational power of a billion desktop-class processors in the power consumption of a single typical handheld device.

Match work to power-efficient option. These approaches are complementary to the preceding approach—energy proportionality—but, rather than having the resources adapt when not fully utilized for a given task, they match tasks to the resources most appropriate to the size of the task. An example is the intelligent use of heterogeneity to improve power efficiency (such as scheduling for asymmetric and heterogeneous multicore processors). Matching work to resources implies there is a choice of resources for a given task. In cluster or multicore environments, the choice exists naturally, but other designs might need to explicitly introduce multiple operation modes with different power-performance trade-offs.

Piggyback or overlap energy events. These approaches seek to combine multiple tasks into a single energy event. For example, multiple reads coalescing on a single disk spin can reduce total disk energy. Prefetching data in predictable access streams or using a shared cache across multiple processes are other examples where such an approach saves energy. Disaggregating or decomposing system functionality into smaller subtasks can help increase the benefits from energy piggybacking by avoiding duplication of energy consumption for similar subtasks across different larger tasks.

Clarify and focus on required functionality. These approaches produce solutions specific to the actual constraints on the design without trying to be too general-purpose or future-proof. For example, special-purpose solutions (such as graphics processors) can be more energy-efficient for their intended workloads. Similarly, designs that seek to provide for future growth by adding modular building blocks can be more energy efficient compared to a single monolithic future-proof design.

Cross layers and broaden the scope of the solution space. Rather than having individual solutions address power management at a local level, focusing on the problem holistically is likely to achieve better efficiencies. Examples

where such an approach have been shown to be effective include scheduling across an ensemble of systems or system components and facilities-aware IT scheduling (such as temperature-aware workload placement). Exchanging information across multiple layers of the networking stack has also been shown to be beneficial for energy efficiency.

Trade off some other metric for energy. These approaches achieve better energy efficiency by marginally compromising some other aspect of desired functionality. An interesting example involves trading off fidelity in image rendering in DVD playback for extended player battery life. Also in this category are optimizations for improved energy delay where improvements in energy consumption significantly outweigh degradations in delay.

Trade off uncommon-case efficiency for common-case efficiency. These approaches seek to improve overall energy efficiency by explicitly allowing degradation in energy efficiency for rare cases and to improve energy efficiency in common cases. For example, a server power supply could be optimized for peak efficiency at normal light loads, even if it leads to degraded power efficiency at infrequent peak loads.

Spend someone else's power. These approaches take a more local view of energy efficiency but at the expense of the energy-efficiency of a different remote system. For example, a complex computation in a battery-constrained mobile device can be offloaded to a remote server in the "cloud," potentially improving the energy efficiency of the mobile device. Approaches that scavenge energy from, say, excess heat or mechanical movement to improve overall energy efficiency also fall in this category.

Spend power to save power. A final category proactively performs tasks that address overall energy efficiency, even though these tasks may themselves consume additional energy. Examples include a garbage collector that periodically reduces the memory footprint to allow memory banks to be switched to lower-power states and a compression algorithm that enables the use of less energy for communication and storage.

The first five categories are well studied and found throughout existing power optimizations. The other five are less common but likely to be important in the future. Combinations are also possible.

Finally, irrespective of which approach is used to improve power efficiency, any solution must include three key architectural elements:

- Rich measurement and monitoring infrastructure;
- Accurate analysis tools and models that predict resource use, identify trends and causal relationships, and provide prescriptive feedback; and
- Control algorithms and policies that leverage the analysis to control power (and heat), ideally coordinated with one another.

From a design point of view, system support is needed at all levels—hardware, software, and application—to facilitate measurement, analysis, control, and cross-layer information sharing and coordination.

Looking Ahead

In spite of all this research and innovation, power management still has a long way to go. By way of illustration, several decades ago, Nobel physicist Richard Feynman estimated that, based on the physical limits on the power costs to information transfer,[5] a staggering 10^{18}-bit operations per second can be achieved for one watt of power consumption. In terms easier to relate to, this implies we should be able to achieve the computational power of a billion desktop-class processors in the power consumption of a single typical handheld device. This is a data point on the theoretical physics of energy consumption, but the bound still points to the tremendous potential for improved energy efficiency in current systems. Furthermore, when going beyond energy consumption in the operation of computing devices to the energy consumption in the supply-and-demand side of the overall IT ecosystem (cradle-to-cradle[3]), the potential is enormous.

The energy efficiency of today's systems can be improved by at least an order of magnitude through systematic examination of their inherent inefficiencies and rethinking of their designs. In particular, in addition to the large body of work in electrical and computer engineering, a new emerging science of power management can play a key role[9] across the broader computer science community. I hope the discussions here—on the design practices that lead to common inefficiencies and the main solution approaches for addressing them—provide a starting framework toward systematically thinking about other new ideas in new domains that will help achieve the improvements. C

References

1. Anderson, D., Dykes, J., and Riedel, E. More than an interface: SCSI vs. ATA. In *Proceedings of the Second Usenix Conference on File and Storage Technologies* (San Francisco, CA, Mar. 31–Apr. 2, 2003), 245–256.
2. Barroso, L.A. and Hölzle, U. The case for energy-proportional computing. *IEEE Computer 40*, 12 (Dec. 2007), 33–37.
3. Chandrakant, P. *Dematerializing the Ecosystem.* Keynote at the Sixth USENIX Conference on File and Storage Technologies (San Jose, CA, Feb. 26–29, 2008); http://www.usenix.org/events/fast08/tech/patel.pdf
4. Cole, G. *Estimating Drive Reliability in Desktop Computers and Consumer Electronics. Tech. Paper TP-338.1.* Seagate Technology, Nov. 2000.
5. Feynman, R. *Feynman Lectures on Computation.* Westview Press, 2000.
6. Irwin, M.J. and Vijaykrishnan, N. Low-power design: From soup to nuts. Tutorial at the International Symposium on Computer Architecture (Vancouver, B.C., June 10–14, 2000); http://www.cse.psu.edu/research/mdl
7. Lefurgy, C., Rajamani, K., Rawson, F., Felter, W., Kistler, M., and Keller, T.W. Energy management for commercial servers. *IEEE Computer 36*, 12 (Dec. 2003), 39–48.
8. Mayo, R.N. and Ranganathan, P. Energy consumption in mobile devices: Why future systems need requirements-aware energy scale-down. In *Proceedings of the Workshop on Power-Aware Computing Systems* (San Diego, CA, 2003), 26–40.
9. National Science Foundation. Workshop on the Science of Power Management (Arlington, VA, Apr. 9–10, 2009); http://scipm.cs.vt.edu/
10. Patel, C. and Ranganathan, P. Enterprise power and cooling: A chip-to-data-center perspective. In *Proceedings of Hot Chips 19* (Palo Alto, CA, Aug. 20, 2007); http://www.hotchips.org/archives/hc19/
11. Rajamani, K., Lefurgy, C., Ghiasi, S., Rubio, J.C., Hanson, H., and Keller, T. Power management for computer systems and datacenters. In *Proceedings of the 13th International Symposium on Low-Power Electronics and Design* (Bangalore, Aug. 11–13, 2008); http://www.islped.org/X2008/Rajamani.pdf
12. Ranganathan, P. (moderator). Power Management from Cores to Data Centers: Where Are We Going to Get the Next 10X? Panel at International Symposium on Low-Power Electronic Devices (Bangalore, 2008); http://www.islped.org/X2008/
13. Rawson, A., Pfleuger, J., and Cader, T. (C. Belady, Ed.). *The Green Grid Data Center Power Efficiency Metrics: Power Usage Effectiveness and DCiE.* The Green Grid, 2007; www.thegreengrid.org
14. Shankland, S. Power could cost more than servers, Google warns. *CNET News* (Dec. 9, 2005); http://news.cnet.com/Power-could-cost-more-than-servers,-Google-warns/2100-1010_3-5988090.html
15. Sullivan, R.F. *Alternating Cold and Hot Aisles Provides More Reliable Cooling for Server Farms.* White paper, Uptime Institute, 2000; http://www.dataclean.com/pdf/AlternColdnew.pdf

Parthasarathy Ranganathan (Partha.Ranganathan@hp.com) is a distinguished technologist in Hewlett-Packard Labs, Palo Alto, CA.