Workloads of the Future

Jan M. Rabaey, Daniel Burke, Ken Lutz, and John Wawrzynek
University of California, Berkeley

Editor's note:
Along with changing technologies and design techniques, target applications span a wide range: from large-scale computing to personal services and perceptual interfaces. The authors of this article characterize these workloads of the future and argue for a new set of benchmarks to guide the exploration and optimization of future systems.
—*William H. Joyner Jr., Semiconductor Research Corp.*

Over the past decade, we have witnessed far-reaching changes in the IT field. Semiconductor sales for consumer and communication devices now surpass those for traditional computation. The IT infrastructure is moving away from the desktop and laptop model to centralized servers, communicating with ubiquitously distributed (and often mobile) access devices. Sensor networks and distributed information-capture devices are fundamentally changing the nature of the Internet from download centric to upload rich (see Figure 1). Whereas today a billion mobile phones are sold per year, in the near future perhaps upwards of a trillion sensory nodes per year will be sold and deployed, with the majority of these connected wirelessly. User interfaces and human-machine interactions could become responsible for a large percentage of the computational needs. This has the potential to fundamentally change the ways we interact with and live in this information-rich world.

This evolution of the IT platform is bound to have a profound impact on the semiconductor business and its operational models. Although Moore's law will still fuel the development of ever more complex devices at lower cost, the nature of these computational and communication devices will probably be substantially different from what we know today, potentially combining hundreds of processing cores. Moving from the core to the fringes of the network, computational prowess will play a less dominant role, and low-power, small form-factor integration of sensors, communication interfaces, and energy sources will be of the essence. It is safe to presume that the "More than Moore" and "Beyond Moore" paradigms will prevail.[1] Where this evolution eventually will lead depends on several factors, such as emerging application needs and the capabilities derived from harnessing the complexity enabled by semiconductor (or its descendant) technologies.

Yet, today a substantial gap exists between the emergent application opportunities and the design community (both hardware and software). New implementation platforms are typically developed in a bottom-up fashion and are largely based on extrapolations of existing applications using old metrics while exploiting technology advances. As a result, these platforms might totally miss the needs and opportunities offered by the nascent applications. In addition, the application community might miscalculate or misinterpret the capabilities of the hardware and software platforms of the future and be lured into dead-ends. Hence, it is in the interest of both communities to meet in the middle by formulating new benchmarks and metrics that better reflect the emerging workloads, thus enabling meaningful design space exploration and system performance analysis.

With all this in mind, this article presents a rough classification of the emergent application areas. From there, we derive a set of specific design metrics that help quantify the effectiveness of candidate implementation platforms. This set is by no means complete, but it underscores the fundamentally diverse nature of emerging IT applications. Our hope is that this discussion will lead to the creation of new benchmark libraries tailored to reflect and measure the properties that are most important to the workloads of the future.

Classifying workloads

For most of us in the IT community, the term *workload* usually refers to *computational workload* and is almost synonymous with the set of traditional computer benchmarks that have been used effectively over the past 30 years to measure and compare the computational effectiveness of, primarily, various

 0740-7475/08/$25.00 Copublished by the IEEE CS and the IEEE CASS

computer architectures; for examples, see the Embedded Microprocessor Benchmark Consortium (http://www.eembc.org) and Standard Performance Evaluation Corporation (http://www.spec.org/benchmarks.html). There have been some efforts recently to expand the scope, an example of which is the recognition, mining, and synthesis (RMS) taxonomy promoted by Intel with commercial and university partners. Our view is that this represents an excellent starting point, but it falls short in some important ways. First, RMS continues to be CPU centric, as is befitting its heritage of anticipating future multicore parallel tasks and tools, whereas our call is for a more device-independent, system-based perspective. Second, the primary evaluation metric remains processor performance, as opposed to broader parameters incorporating energy, latency, and reliability, among others.

Although general-purpose computing will continue to demand a sizable (if not dominant) fraction of the design complexity, power consumption, and system cost, a birds-eye view of the IT landscape reveals some major emerging trends that require some fundamentally different testbenches and metrics. Based on a perusal of the different information-based industries, we have derived the following four property-based classifications.

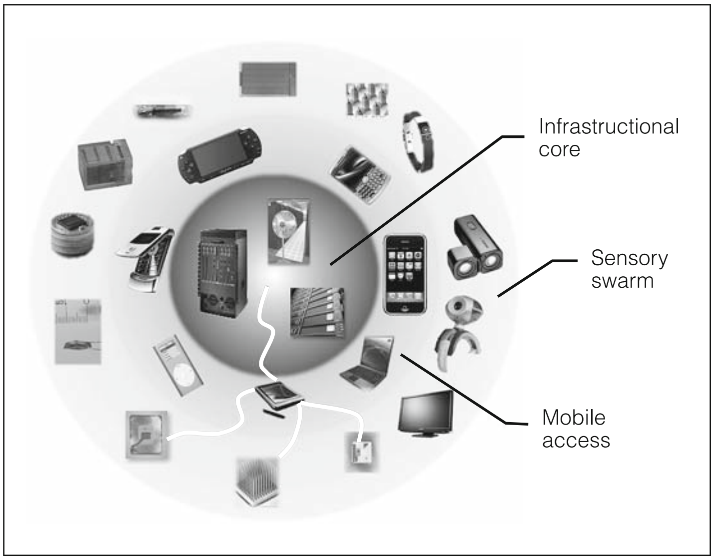

Figure 1. The evolving information technology scene.

High-performance computing tasks

There is no doubt that traditional high-performance computation will continue its explosive growth and its insatiable demand for more computing cycles. Complex scientific problems in fields such as climate research, material science, chemistry, particle physics, and life sciences continue to be the driving applications. The search for representative benchmark sets has been a continuing effort, an example of which is the National Energy Research Scientific Computing Center (NERSC) Sustained System Performance (SSP) benchmark set.[2] Yet even in this community, the emergence of massively parallel embedded-processing platforms (some of which have arisen from unexpected corners such as graphics processing) has forced a rethinking of how to best capture applications, of what algorithms have the best scaling properties, and of which metrics to apply. Power-consumption concerns increasingly limit the computational throughput that can be delivered by high-performance computing systems.

At the same time, the emergence of massively parallel data and storage centers, fueled both by the rampant growth of the Internet and by centralized data services, has given rise to a new perspective on the properties and needs of centralized computation. Some applications, such as Internet search, are embarrassingly parallel; isolating millions of (small) simultaneous threads is not a problem. Instead, latency, reliability, and power-dissipation guarantees have emerged as the prime concerns for these applications classes.

An often forgotten, yet essential component behind these advances is the capability of communicating huge amounts of data either over short (within the data center) or long (over the Internet) distances, with wireless interconnects becoming a more important medium at the fringes of the network. This has led to *virtualized computing resources*—that is, computational power can be swapped in and migrated at will. Power dissipation of network routers has become a sizable fraction of the power consumed in data centers and the Internet network infrastructure.

Complex distributed systems

When speculating about the future, it seems that the issues of engineering efficient computation and communication infrastructures will be overshadowed by the challenges presented by the growth of what has often been called a *societal IT system* (SIS). The Center for Information Technology Research in the Interest of Society at the University of California, Berkeley originally coined the name in 2001 (see http://www.citris-uc.org). In IBM terminology, this is referred to as

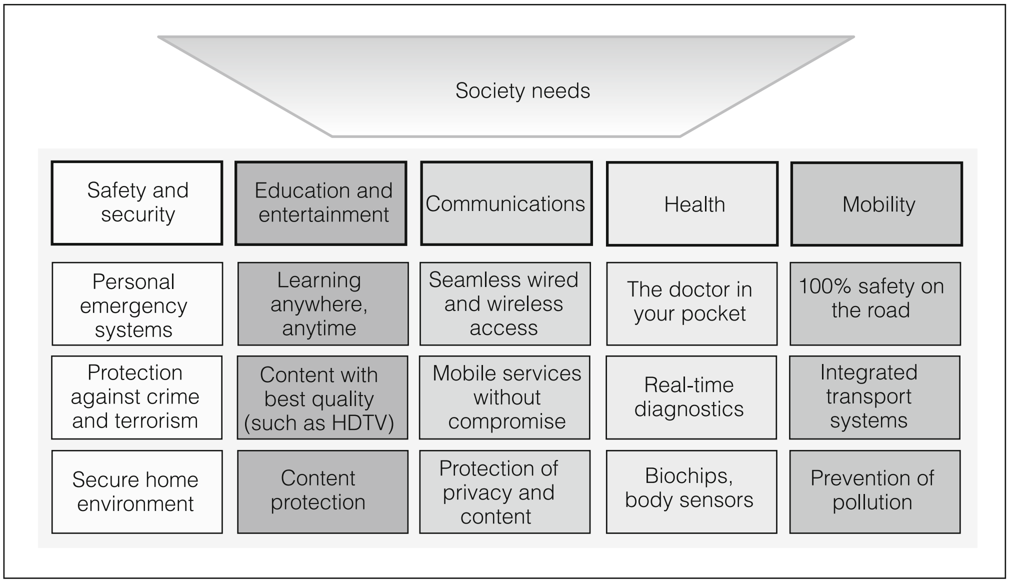

Figure 2. Societal needs and applications. (Source: ENIAC Strategic Research Agenda; courtesy of Johan Janssen, NXP.)

world-aware computing. A SIS generally consists of a distributed system of sensors, computing and control devices, and actuators that work together to address several major problems affecting our daily lives. Examples of such problems include automotive or avionic safety, traffic-flow management, environmental control and safety protection in *high-performance buildings* (often called smart homes), distributed health monitoring, and power distribution with decentralized energy generation (see Figure 2). The scale of such systems can vary widely, from a small locale (within a single automobile, for instance) to a residential dwelling, a metropolitan area, or even a nation-wide or worldwide setting. All these systems share some common properties: they are complex; often exhibit emergent behavior; and must be fail-safe, scalable, and flexible. Generally, all such systems can be characterized as having many inputs and outputs, requiring distributed computing, and being power constrained.

For example, Figure 3 illustrates the main characteristics behind the high-performance building concept. The challenges here are that today's systems are fragile and nonscalable, and they lack the flexibility to adapt to changing conditions. Enabling technologies that can help meet these challenges are the rising availability of ubiquitous and redundant sensors, always-connected wireless networks, high-performance data aggregation, information-context extraction, and distributed processing.

The concept of always-connected wireless networks deserves some special attention. Today's wireless networks rightfully have the reputation of being unreliable and of providing spotty coverage. Yet, some nascent ideas could fundamentally change this situation. The exploration of novel spectral bands combined with far better utilization of existing bands using intelligent cognitive radios could remove some of the capacity limitations.[3] These approaches could be combined with *collaborative networks*, where wireless nodes work together to ensure connectivity even in the event of failed nodes or a failed infrastructure (using mesh networks, for example), and where connectivity brokerage offers an incentive for nodes to collaborate.

These ideas will conspire to fundamentally change the way we connect and communicate, and they could lead to a perceived unlimited bandwidth. In

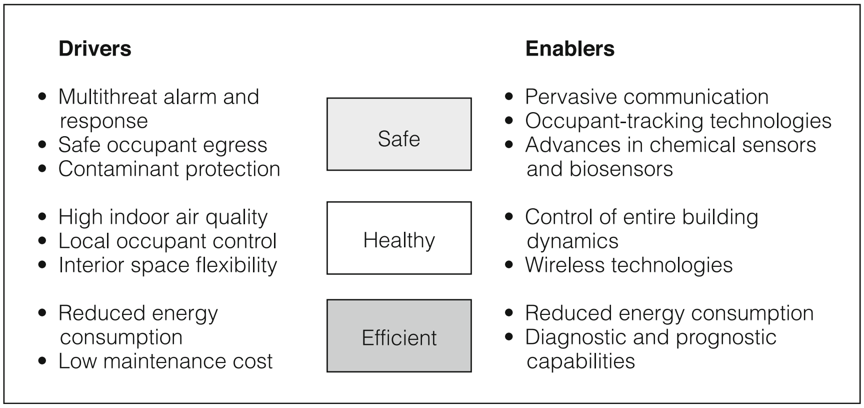

Figure 3. High-performance buildings: drivers and enablers. High-performance buildings are those that meet or exceed the needs of the occupants at lower life cycle costs and resource utilization than typically realized using conventional methods. (Courtesy of Clas Jacobson, United Technologies Corp.)

fact, always-connected wireless networks have all the properties of a SIS. The realization of such an operational SIS puts various stringent demands on the semiconductor industry, some of which diverge from the traditional technology-scaling model. The primary challenges will be meeting the reliability requirements, managing the complexity of these systems, aggressively reducing their power consumption, and of course, continuing miniaturization and integration, often including widely different technologies. Reliability and complexity management don't necessarily have to be addressed at the IC level, because system-level strategies could be just as effective. It is essential, however, that all components in a SIS present a composable model that enables seamless integration of elements from a range of vendors. In addition, components should be self-checking (and self-correcting, if possible), letting the system reconfigure in case of failure.

Personalized services

The complexity of such a SIS is daunting to even the most expert user. Although it is possible to gather voluminous data with a SIS, presenting only the relevant information in the right form and at the right time is essential. Hence, an ever-growing fraction of the IT industry is pursuing the business of providing personalized services. Even now, when pulling up a personalized web page, an elaborate set of actions is put into motion, collecting information from many sources using profile-based information and assembling it into a single page, all within a short time span. For example, the occupant of a residence or office in a high-performance building does not care about the electronic system's internal architecture and does not want to worry about its maintenance.

To address these concerns, a uniform set of services layered atop the SIS can provide ease of use to deliver relevant information when needed. In one application, a John Deere tractor driving over an agricultural field could forward measurements of soil PH and moisture to a data center that performs fertilization optimization and returns commands to the dispensing system in a closed loop.[4] These type of services require the combination and integration of many diverse components.

Perhaps the most important challenge is managing latency. Users expect and require a fast response, even though that response might require communication over long distances, extensive computation, and complicated data mining. Reducing latency is mostly an issue of system trade-offs, which include the decision of where and when to compute. Adaptive distribution of the computational tasks is critical.

Perceptual processing

A final but essential component is the evolution toward more advanced user interfaces. *Perceptual processing* is data processing that exploits the properties and limitations of human perception to simplify a computation while maintaining the perceived quality.

The way we interact with information systems has not fundamentally changed over the past four decades, but the amount of data we input and must process has grown exponentially. Tim Mattson from Intel explains, "We must go beyond batch and interactive interfaces; [the interface] must immerse the human into the computation."[5] Again, the ubiquitous availability of miniature sensor nodes would make it possible to augment the available senses and provide a much broader bandwidth between humans and the computing environment surrounding them.

Consider the Nintendo Wii gaming console. Although it does not by any means provide the best graphics or the highest computational performance, the immersive experience provided by an accelerometer-based user interface has made this platform an instant success. Similarly, even in quasi-saturated markets, novel interfaces can make all the difference—the Apple iPod and iPhone are good examples. Yet, these are just the beginning. Adding voice and visual inputs will make these mobile devices more instinctive and effective. Including other sensory inputs such as motion or physiological measurements, combined with contextual information, will create a fundamentally different experience.

For immersive interfaces to be successful, they often require enormous amounts of processing, including recognition, classification, rendering, and synthesis. The immersive computational pipeline in Figure 4 illustrates some of this. Providing computational power of this magnitude in a mobile device (with severely constrained energy) seems impossible. But two considerations can help mitigate these concerns:

- the availability of an always-connected low-latency wireless network makes it possible to move some, if not most, of the computational load to the backbone; and

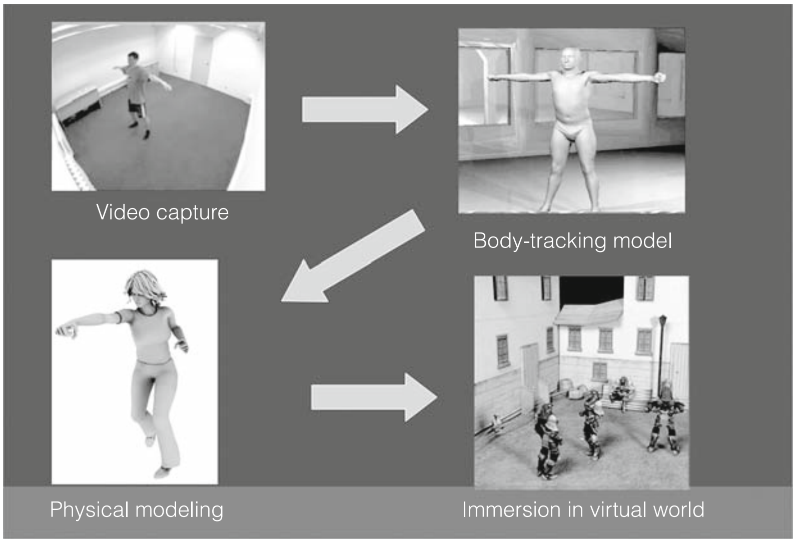

Figure 4. The immersive computational pipeline. (Courtesy of Tim Mattson, Intel.)

- many of the computations involved in perceptual processing are error tolerant, which opens the door for the development of platforms that are far more forgiving to errors.

We may even consider using computational models that fall out of the traditional Boole-Van Neumann-Turing model and might map far more effectively on some of the emerging nanoplatforms or bioplatforms. An excellent starting point to explore implementation platforms for immersive interfaces is the Princeton Application Repository for Shared Memory Computers application benchmark set (http://parsec.cs.princeton.edu), which captures conventional RMS benchmarks and representatives of emerging large-scale multithreaded commercial programs.

Metrics redefined

In the past, we tended to focus on just a few simple measures. Considering the range of applications we have outlined thus far, however, it is clear that new metrics must apply when judging how well a proposed implementation platform correlates with a particular application.

Traditional metrics

In the traditional computer architecture arena, raw performance has long been the target. Several metrics to quantify performance have thus come into vogue. The execution of MIPS (million instructions per second), while often deceptive, has been the most popular. A more precise metric is measuring the time to execute a program, which is the product of the instructions in the program, the number of instructions per cycle, and the cycle time.[6] Obviously, quantifying this number requires the availability of a representative collection of applications. Other metrics have been proposed, but the goal of all of them is ultimately the same: expressing raw computational throughput.

With power becoming an issue over the past decade, a second set of metrics emerged measuring a computational platform's energy efficiency. The most popular metric is to measure how much energy it takes on average to perform an instruction. The inverse metric (how many instructions can be performed for a given amount of energy) is often used as well. The average power the processor consumes is then computed by multiplying the energy per instruction by the number of instructions per cycle and the cycle frequency.

In light of the emerging workloads, these metrics either only reflect a small part of the story or are rendered irrelevant. Hence, a broad analysis of important new metrics is essential and long overdue.

Future workload metrics

One picture that clearly emerges from our analysis of the application spaces is that raw performance, while still important, is not at the top of what is deemed essential for an implementation platform to be a good match to an application. Just-enough performance is often just fine. In a world where applications are performed on concurrent and distributed platforms, other qualitative measures apply.

Useful functionality and energy. Usage of mobile distributed components undoubtedly represents the largest growth factor in the IT world over the coming years. Hence, power and energy efficiency will likely be some of the most compelling metrics, and dramatic improvements in energy efficiency will be needed if some of the proposed scenarios are to become reality. Example workloads where these metrics are essential include those involving handheld devices such as smart phones. Energy efficiency is downright critical in the case of self-contained embedded sensor nodes (for instance, in medical implants or intelligent environments such as high-performance buildings). Somewhat surprisingly, efficiency has also become of prime importance in data centers, where the power bill represents the dominant operational cost.

Yet, a straightforward energy-efficiency metric does not do justice to the reality of the distributed world. In a connected world, a single task combines local processing, communication, and remote computation. The total energy needed to execute the overall task does not really matter much to mobile users; what counts is the quality of their perceived overall experience for the energy spent on the mobile device. John Shen from Nokia explains, "The important metric to optimize is 'user experience per unit energy.'"[7]

Of course, "user experience" is a qualitative measure, but it can be quantified for certain attributes such as quality of service (QoS) or total hours of connect time for a single battery charge. This *user experience per unit energy* metric effectively decouples global computational requirements from local energy consumption. Moving functions to the backbone network (thereby trading local computation for communication) might effectively reduce local energy consumption or adversely affect other quality metrics such as latency.

Sometimes communication trade-offs are not directly evident. Consider, for example, the case of multihop wireless mesh networking. By relaying messages for other users, it might seem that users would be penalizing their own experience. However, as it turns out, relaying messages on behalf of others can result in system-wide energy efficiency, with each user transmitting more data for the same battery charge. Other energy system-level metrics might be relevant as well. For example, in distributed sensor networks, it's often important to optimize the network's lifetime, given the available energy stored at the participating nodes. The network's lifetime is another QoS metric, which we might define as the time until a catastrophic failure occurs or the network performance degrades to a certain point.

System latency. In any task that is life critical or in which a human is in the loop, meeting end-to-end latency constraints is essential. If constructing web pages on the fly, based on personality profiles or traffic updates to a mobile user, exceeds some latency constraints (that is, the user's patience), the application is doomed. Excessive latency might make immersive computing a highly unpleasant or even sickening experience, as researchers (and companies) in the world of immersive user interfaces quickly figured out. Similarly, in a high-performance building, the networking delay might make the best sensoring system totally useless. As Clas Jacobson from United Technologies Corp. put it, "smoke travels fasters than bits." Even in high-performance throughput-oriented computing, latency is becoming one of the essential measures. With the emerging massively parallel many-core processors, it is rare for instantaneous computational bandwidth alone to determine the actual execution time for a given task.

In a distributed system, many components contribute to latency, a large fraction of which have little to do with computational power, or sometimes even communication speed. As with the user experience and energy metric, optimization often requires exploring different system architectures and adaptively relocating functionality. For example, if a mobile user's location is known, it's possible to prefetch relevant data to either the mobile device or data servers close to its location in order to substantially reduce latency.

Reliability and liability. Although reliability has always been a concern in IT system design, it is absolutely at the forefront in a SIS. This is best illustrated with the examples of some of our projected workloads. Avionics and automotive systems must be absolutely fail-safe. Failure or instability in a high-performance building's safety systems could be life-threatening. The financial implications of a failed metropolitan power-distribution system are enormous. Unfortunately, wireless communication systems often fail during emergencies, when they are most needed. A common property in all these scenarios is that the lack of reliability directly translates into financial liability.

This quest for absolute reliability is gaining momentum just at a time when the underlying hardware platforms are becoming increasingly unreliable. Several reasons account for this:

- scaling of semiconductor technology to the nanometer scale,
- the increasing complexity of system components and of the systems themselves, and
- the distributed nature of most systems.

At the same time, some of the workloads can sustain a certain level of failure or uncertainty without being affected. Perceptual processing is a perfect example. Most human-machine interface functions are highly based on subjective interpretation, and as long as the results are within range, they are perfectly acceptable. Even better, when closed feedback is involved, the entire human-computer system often adapts to bring the results to the desired operational point. This realization can have a profound impact on the computation platforms used for perceptual processing. Building highly reliable systems (or systems with just the right amount of reliability) requires a culture and, increasingly, a structured design methodology in which reliability is treated as a key metric. In such a design methodology, quantifiable measures of reliability must be present at all levels of the design hierarchy.

A critical realization is that reliability in a distributed system does not necessarily mean that the individual components or links must be absolutely bulletproof. In fact, the system reliability is a statistical property, which results from the combination of the statistics of the individual components. Thanks to information redundancy, it is possible to create systems that are ultimately reliable even in the presence of degrading or failing components or links.

Complexity, modularity, and composability. Complexity and composability are most often considered a system property (or component thereof), not a metric. Yet, in a world where large systems are constructed by assembling heterogeneous elements produced by many different vendors under ever shorter time-to-market constraints, composability with functionality, performance, and reliability guarantees is essential. Although this challenge has always existed, it is becoming far more pronounced in a business climate in which companies are more horizontally structured.

Similar trends are occurring in various industries, ranging from building SoCs using IP components to assembling cars using components from a range of suppliers. System designers should have the analytical tools in hand to quickly explore design options, judge risk, and assess opportunities. Today, composability metrics do not exist. Most probably, a set of weighting functions could be derived to translate today's qualitative understandings into a well-defined metric. Doing so is (and should be) the topic of intense research.

Although information technology has transformed itself many times over the past decades, a new and fundamental overhaul is in the making. The long-predicted world of fully ubiquitous computation and communication is finally emerging, bringing with it a whole new set of applications, platforms, challenges, and questions. To maximally exploit the offered opportunities, it is essential that a new set of benchmark libraries be developed to inform the design and exploration process. These benchmarks must reflect the properties we have outlined here, because they are fundamental to the workloads of the future. Creating benchmarks is more challenging now than in the past, primarily because the new workloads go beyond the single-component level of old and extend to the distributed-system level, where integration issues create the greatest difficulties.

Formulating these new benchmarks and related metrics that better reflect the emerging workloads is essential if meaningful design space exploration and system performance analysis are to take place. The success of future system design technologies hinges on these efforts. Consequently, it is imperative that all the communities involved start engaging without further delay. ■

Acknowledgments

Many, if not most, of the ideas we present in this article emerged from a GSRC workshop titled ''Workloads of the Future,'' held in Santa Clara, California, on 30 November 2007. We acknowledge the insightful and forward-looking contributions of the workshop speakers and session chairs: Ranveer Chandra, Brian Cooper, Greg Heinzinger, Wen-mei Hwu, Clas Jacobson, Johan Janssen, Tim Mattson, Patrick Le Gresley, Massimo Osella, Ammar Rayes, Jim Rowson, Alberto Sangiovanni-Vincentelli, John Shalf, and John Shen. We also acknowledge the support of the Gigascale Systems Research Center (GSRC) and the Focus Center Research Program (FCRP) Consortium.

■ References

1. H. De Man, "Ambient Intelligence: Gigascale Dreams and Nanoscale Realities," *Proc. IEEE Int'l Solid-State Circuits Conf.* (ISSCC 05), IEEE Press, 2005, pp. 29-35.
2. W.T.C. Kramer, J.M. Shalf, and E. Strohmaier, "The Sustained System Performance (SSP) Benchmark," http://www.nersc.gov/projects/ssp.php.

3. D. Čabrić et al., "A Cognitive Radio Approach for Usage of Virtual Unlicensed Spectrum," *Proc. 14th IST Mobile and Wireless Communications Summit*, IEEE Press, 2005, http://www.eurasip.org/Proceedings/Ext/IST05/papers_A.html.
4. D. Rickman et al., "Precision Agriculture: Changing the Face of Farming," *GeoTimes*, Nov. 2003, http://www.geotimes.org/nov03/feature_agric.html.
5. T. Mattson et al., "Panel: Is the Multi-Core Roadmap Going to Live Up to Its Promises?" *Proc. IEEE Parallel and Distributed Processing Symp.* (IPDPS 2007), IEEE Press, 2007, p. 14.
6. D.A. Patterson and J.L. Hennessy, *Computer Organization and Design: The Hardware/Software Interface,* 3rd ed., Morgan Kaufman, 2007.
7. J.P. Shen, "Mobility, Community, Serendipity," *GSRC-FCRP Workshop: Workloads of the Future,* 2007.

The biography of **Jan M. Rabaey** is on p. 302 of this issue.

Daniel Burke is a research engineer in the Berkeley Wireless Research Center at the University of California, Berkeley. His research interests include reconfigurable computing, embedded systems and software, and design for reliability. He has a BS in physics from Texas A&M University and an MSEE in electrical and computer engineering from the University of Illinois at Urbana-Champaign. He is a member of the IEEE.

Ken Lutz is the executive director of the Gigascale Systems Research Center (GSRC) at the University of California, Berkeley. His research interests include large-scale systems design and networking. He has a BS in electrical engineering from the University of California, Davis. He is a member of the IEEE.

John Wawrzynek is a professor in the Electrical Engineering and Computer Science Department at the University of California, Berkeley. His research interests include computer architecture, reconfigurable computing, and integrated circuits and systems design. He has a BS in electrical engineering from the State University of New York, Buffalo, an MS in electrical engineering from the University of Illinois at Urbana-Champaign, and a PhD in computer science from the California Institute of Technology.

■ Direct questions and comments about this article to Jan M. Rabaey, Berkeley Wireless Research Center (BWRC), University of California, Berkeley, 2108 Allston Way, Suite 200, Berkeley, CA 94704; jan@eecs.berkeley.edu.