Workloads of the Future

Jan M. Rabaey, Dan Burke, Ken Lutz and John Wawrzynek

Berkeley Wireless Research Center (BWRC)
University of California, 2108 Allston Way – Suite 200
Berkeley, CA94704

{jan@eecs.berkeley.edu, drburke@bwrc.eecs.berkeley.edu, lutz@eecs.berkeley.edu, johnw@cs.berkeley.edu}
tel: (510) 666 3102 (for all authors)
fax: (510) 883-0270

Abstract

Information technology has transformed itself many times over the past decades. Yet a far more fundamental overhaul is in the making – the long-predicted world of fully ubiquitous computation and communication is finally emerging. Addressing the associated challenges and opportunities may require us to fundamentally re-think the way we do design. Most notably, the semiconductor industry must abandon its traditional “component-oriented” perspective and adopt a system vision. To meaningfully guide this process and to fully exploit the offered opportunities, an understanding of what the workloads of the future may look like is necessary, a task which can only be accomplished through a joint effort of the application and design communities.

This paper characterizes the essential features of what we believe to be dominant application classes of the future. It becomes apparent that these workloads operate under different quality metrics. Instead of performance or energy per function, metrics such as system latency, useful functionality per energy spent, and reliability/liability take center stage. We therefore issue a call to action for the creation of relevant benchmark libraries, each accompanied with a clear definition of the metrics relevant to their evaluation.

Keywords
Ubiquitous computing, distributed computing, benchmarks, metrics, system-level design.

Author Biographies

Jan Rabaey is the Donald O. Pederson Distinguished Professor at the EECS Department of the University of California at Berkeley. His current research interests include the conception and implementation of next-generation integrated wireless systems. He received the Ph.D. degree in Applied Sciences from the Katholieke Universiteit Leuven, Belgium. He is a Fellow of the IEEE, and director of the GSRC focus center.

Daniel Burke is a research engineer at 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. Burke has a BS in physics from Texas A&M, and an MSEE in electrical and computer engineering from the University of Illinois, Urbana-Champaign. He is a member of the IEEE.

Ken Lutz is the Executive Director of the Gigascale Systems Research Center (GSRC) headquartered at the University of California, Berkeley. His research interests include large-scale systems design and networking. Ken received his BSEE from the University of California, Davis in 1979. He is a member of the IEEE.

John Wawrzynek is a Professor at the EECS Dept of the University of California at Berkeley. His research interests are in computer architecture, reconfigurable computing, and integrated circuits and systems design. He received a B.S. in Electrical Engineering from SUNY, Buffalo, 1977, a M.S. in EE from the University of Illinois, Urbana/Champaign, 1979, and a Ph.D. in Computer Science from the California Institute of Technology, 1987.
A. Introduction

Over the past decade we have been witness to far-reaching changes in the information-technology (IT) field. Semiconductor sales for consumer and communication devices now surpass those for traditional computation. The information-technology infrastructure is moving away from the desktop/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. Where today a billion mobile phones are sold per year, in the not so distant future, perhaps upwards of a trillion sensory nodes per year will be sold and deployed—-with the majority of these being wirelessly connected. This has the potential to fundamentally change the ways we interact with and live in this information-rich world. User interfaces and man-machine interactions may be responsible for a large percentage of the computational needs in this future. In short, information technology is in turmoil.

Figure 1: The evolving information-technology scene.

This evolution of the information-technology platform is bound to have a profound impact on the semiconductor business and its operational models. While Moore’s law will still fuel the development of ever more complex devices at lower cost, the nature of these computational/communication devices will in all probability 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 plays less and less of a dominant role, and low-power, small form-factor integration of sensors, communication interfaces, and energy sources is of the essence. It is safe to presume that the
“More than Moore” and “Beyond Moore” paradigms$^1$ prevail here. Where this evolution eventually will lead depends on a number of factors, such as emerging application needs as well as the capabilities from harnessing the complexity enabled by the semiconductor (or its descendant) technologies.

Yet today a substantial gap exists between the emergent application opportunities and the design community (both hard- and software). New implementation platforms are typically developed in a bottom-up fashion, and are largely based on extrapolations of existing applications using the metrics-of-old while exploiting technology advances. As a result, these platforms may totally miss the needs and the opportunities offered by the nascent applications. In addition, the application community may miscalculate or misinterpret the capabilities of the hardware/software platforms of the future, and be lured in dead-end directions. 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, hence enabling meaningful design space exploration and system performance analysis. Doing so may indeed lead to some entirely innovative and surprising solutions!

We therefore undertake an attempt to present a rough classification of the emergent application areas. From this, 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 underscores the fundamentally different nature of the emerging IT applications. Our true hope is that the outcome of this may be the creation of new benchmark libraries tailored to reflect and measure the properties that are most important to the workloads of the future.

B. Perceived Needs – Classification of Workloads

For most of us in the information-technology 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 years to measure and compare the computational effectiveness of, primarily, various computer architectures$^{2,3}$. There have been some efforts recently to expand the scope, an example of which is RMS taxonomy promoted by Intel with commercial and university partners. Our view is that this represents and excellent starting point, which however falls short in several 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.

While 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 information-technology landscape reveals some major emerging trends, which require some fundamentally different test-benches and metrics.

Based on a perusal of the different information-based industries, we derived the following “property-based” classifications.
1. High-Performance Computing Tasks

There is no doubt that traditional high-performance computation will continue its explosive growth and its demand for ever more compute 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 NERSC “Sustained System Performance (SSP)” benchmark set. Yet even in this community, the emergence of massively parallel embedded processing platforms (some of which arose 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 totally 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 distances (over the Internet), with wireless interconnects becoming an ever more important medium at the fringes of the network. This has led to the concept of “virtualized compute 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 Internet network infrastructure.

2. Complex Distributed Systems

When reflecting 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 often have been called Societal Information-Technology Systems (SIS). (In IBM terminology, this is referred to as world-aware computing). An SIS in general consists of a distributed system of sensors, compute and control devices and actuators that work together to address a number of major problems affecting our daily lives. These problems, for instance, could be automotive or avionic safety, management of traffic flows, environmental control and safety protection in “high-performance buildings”, distributed health monitoring, and power-distribution with decentralized energy generation (Figure 2). The scale of such systems can vary widely, from a small locale (within a single automobile, for instance), to a residential dwelling or a metropolitan area, and possibly even to a nation- or world-wide setting. All of these systems share some common properties: they are complex, they often exhibit emergent behavior, and they must be “fail-safe”, scalable and flexible. In general, all such systems can
be characterized as having large numbers of 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 non-scalable, and they lack the flexibility to adapt to changing conditions. Enabling technologies to 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.
“collaborative networks”, where wireless nodes work together to ensure connectivity even in the event of failed nodes or infrastructure (mesh networks are an example of such), and where connectivity brokerage offers an incentive for nodes to collaborate. These ideas conspire to fundamentally change the way we may connect and communicate in the future, and may lead to a perceived unlimited bandwidth. In fact, always-connected wireless networks have all the properties of an SIS themselves.

The realization of operational SIS systems puts a number of 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 do not necessarily have to be addressed at the integrated-circuit level, as system-level strategies could be just as effective. It is essential, however, that all components in an SIS system present a compose-able model that allows elements from a wide range of vendors to be put together seamlessly. In addition, components should be self-checking (and correcting if possible), allowing the system to reconfigure in case of failure.

3. **Personalized Services**

The complexity of these SIS systems is daunting to even the most expert user. While with these systems it is possible to gather voluminous data, presenting only the relevant information in the right form and at the right time is essential. Hence, an ever-growing fraction of the information-technology 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 them into a single page, all within a very short time span. The occupant of either a residence or office that is equipped as a high-performance building does not care about the internal architecture of the electronic system and does not want to worry about its maintenance. To address these concerns, a uniform set of services layered on top of the SIS can provide ease of use and to deliver relevant information when needed. Similarly, a John Deere tractor driving over an agricultural field can forward measurements of soil PH and moisture to a data center which performs fertilization optimization returning commands to the dispensing system in a closed loop. Services of these types require the combination and integration of many diverse components. Perhaps the most important challenge is the management of latency. Users expect and require fast response, even though the delivery of that response may 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 of the essence.

4. **Perceptual Processing**

A final but essential component of the picture is the evolution towards more advanced user interfaces. The way we interact with information systems has not fundamentally changed over the past decades. Yet the amount of data we input and must process has grown exponentially. Quoting Tim Mattson from Intel: “We must go beyond batch and interactive interfaces; it [the interface] must immerse the human
into the computation”. Again, the ubiquitous availability of miniature sensor nodes makes it possible to augment the available senses, and to provide a much broader bandwidth between the human and the compute environment surrounding it. Consider, for example, the Nintendo Wii gaming console. While not by any means providing the best graphics or the highest computational performance, the immersive experience provided by an accelerometer-based UI has made this platform an instantaneous success. Similarly, even in quasi-saturated markets, novel interfaces can make all the difference—the Apple iPod and iPhone are good examples. Yet these just represent a start. Adding voice and/or visual inputs is one step to make these mobile devices more instinctive and effective. Including other sensory inputs such as motion or physiological measurements, combined with contextual information, would create a fundamentally different experience.

For immersive interfaces to be successful, they often require enormous amounts of processing, including recognition, classification, rendering and synthesis. Some of this is illustrated in Figure 4, which represents the immersive computational pipeline. Providing computational power of this magnitude in a mobile device (with severely constrained energy) seems to be impossible. Two considerations can help mitigate these concerns: (1) 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 (2) many of the computations involved in “perceptual computing” are error-tolerant—opening the door for development of platforms that are far more forgiving in nature. One may even consider the use of computational models that fall out of the traditional Boole-Van Neumann-Turing model and that may map far more effectively on some of the emerging nano- or bio-platforms.

An excellent starting point to explore implementation platforms for immersive interfaces is the application benchmark set PARSEC (Princeton Application Repository for Shared memory Computers), which additionally captures conventional recognition, mining, and synthesis (RMS) benchmarks as well as representatives of emerging large-scale multi-threaded commercial programs.

Figure 4: The immersive computational pipeline
(Courtesy: Tim Mattson, Intel)
C. Metrics Redefined

Considering the broad range of applications outlined in the previous sections, it is clear that new metrics must apply when judging how well a proposed implementation platform is matched to a particular application. In the past, we tended to focus on just a few simple measures.

a. The Traditional Metrics

In the traditional computer architecture arena, raw performance has long been the target of choice. A number of metrics to quantify performance have hence come into vogue. The MIPS (Millions of Instructions executed per Second), while often deceptive, has been the most popular one. A more precise metric is Time to Execute a Program (the product of the instructions in the program, the number of instructions per cycle, and the cycle time)\(^1\). 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 last decade, a second set of metrics emerged measuring the energy “efficiency” of a computational platform. The most popular metric is energy/instruction, which measures how much energy it takes on average to perform an instruction. The inverse metric (that is, how many instructions can be performed for a given amount of energy) is often used as well. The average power consumed by the processor 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.

b. The Metrics of Future Workloads

One picture that clearly emerges from the analysis of the application spaces discussed earlier 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. A set of potential metrics, broadly applicable, is elaborated below.

Useful functionality / energy

Usage of mobile distributed components undoubtedly represents the largest growth factor in the information technology world over the coming years. Hence one would expect that power and energy efficiency are some of the most compelling metrics. And indeed, dramatic improvements in energy efficiency are needed if some of the proposed scenarios are to become reality. Example workloads where this metric is of essence 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, as used 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 factor.

Yet, a straightforward energy-efficiency metric does not do the reality of the distributed world justice. 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 the mobile user – what counts is the quality of the overall experience she or he perceives for the energy spent on the mobile device! To quote John Shen from Nokia: “The important metric to optimize is ‘user experience per unit energy’. Of course, “user experience” is generally a qualitative term, however, it can be quantified for certain attributes, such as obtained quality-of-service, or total hours of connect time. This “user experience per unit energy” metric effectively decouples global computational requirements from local energy consumption. Moving functions to the backbone, trading local computation for communication, may have the effect of reducing local energy consumption, however, it might adversely affect other quality metrics such as latency (as discussed below). Sometimes communication trade-offs are not directly evident. Consider for instance the case of multi-hop wireless mesh networking. By relaying messages for other users, it may seem that a user would be penalizing their own experience. However, as it turns out, relaying messages on behalf of others, can result in systemwide energy efficiency with each user transmitting more data for the same battery charge.

Other energy “system levels” metrics may be relevant as well. For instance, in distributed sensor networks it is often important to optimize the lifetime of the network given the available energy stored at the participating nodes. The lifetime of the network is another QOS metric, definition of which could be the time until catastrophic failure occurs, or the network performance degrades to a certain point.

System Latency

In any task that is life critical or where a human is in the loop, meeting end-to-end latency constraints is essential. If on-the-fly construction of web pages based on personality profiles or traffic updates to a mobile user exceeds some latency constraints (that is, the patience of the user), the application is doomed. Excessive latency may 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 may make the best sensing system totally useless. To quote Clas Jacobson from UTC, “smoke travels faster than bits”. It is interesting to observe that 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 that instantaneous computational bandwidth alone determines the actual execution time for a given task.

In a distributed system, many components contribute to latency, a large fraction of which has little to do with computational power, or sometimes even communication speed. As with the “user experience/energy” metric, optimization often requires exploring different system architectures and
adaptive re-location of functionality. For instance, if the location of a mobile user is known, it is possible to pre-fetch relevant data to either the mobile device or to data servers close to its location, hence substantially reducing the latency.

Reliability/Liability

While reliability has always been of concern in the design of information-technology systems, it is absolutely at the forefront in societal-scale information systems. This is best illustrated with the examples of some our projected workloads. Avionics and automotive systems must be absolutely fail-safe. Failure or instability in the safety systems of a high-performance building can be life threatening. The financial implications of a failed metropolitan power-distribution system are enormous. Unfortunately, wireless communication systems often fail when they are most needed—that is, in the case of emergencies. A common property in all these scenarios is that 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 the latter: scaling of semiconductor technology to the nano-meter scale, the increasing complexity of system components and of the systems themselves, and finally 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 in the outcome. Perceptual processing is a perfect example of such. 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, even more, a structured design methodology in which reliability is treated as key metric. In such a design methodology, quantifiable measures of reliability must be present at all levels of the design hierarchy. A crucial 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. Due to redundancy in the information, it is possible to create systems that are ultimately reliable even in the presence of degrading or failing components or links.

Complexity, Modularity, Composability

Complexity and composability are most often considered a property of a system (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. While this challenge has always existed, it is becoming far more pronounced in a business climate where companies are more and more horizontally structured. Similar trends are occurring in different
industries, ranging from building systems-of-a-chip from IP components to assembling cars from components from a broad range of suppliers. System designers should have the analytical tools in hand to quickly explore design options, judge risk, and assess opportunities. Today, these metrics of composability do not exist. Most probably, a set of weighting functions can be derived to translate today’s qualitative understandings into a well-defined metric. Doing so is (and should be) the topic of intense research.

D. Call To Action

While 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 in the paper, as they are fundamental to the workloads of the future. The creation of benchmarks is more challenging now than in the past. The single most compelling reason for this is that that the new workloads go beyond the single component level of old, and extend to the distributed system level, where integration issues create the biggest difficulties. The formulation of these new benchmarks and related metrics that better reflect the emerging workloads is essential if meaningful design space exploration and system performance analysis is to take place.

The success of future system-design technologies hinges on these efforts. In consequence, it is essential that all the communities involved start engaging without further delay.

Acknowledgements

Many if not most of the ideas presented in this paper emerged from a GSRC workshop named “Workloads of the Future”, held in Santa Clara, CA on November 30, 2007. The authors 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. The support of the FCRP consortium for this effort is greatly appreciated.

References


5 Term originally coined by CITRIS UCB Berkeley, 2001: http://www.citris-uc.org


10 PARSEC, http://parsec.cs.princeton.edu/
