Participant Inputs in Area 4: Systems, Applications and Beyond

Applications:

1. What will be the dominant workload (i.e., computational load and memory/network traffic pattern) for different application classes in 5-10 years?

Answers:

I believe the dominant workload and network traffic pattern will be

- multimedia: high-def content with interactive control and volumetric data (3D, predictable, high bandwidth, somewhat latency tolerant)
- cloud-driven data synchronization with fine-grained merging of changes (unpredictable bandwidth, not latency tolerant)
- cloud support for wireless sensing and actuation networks e.g., home/grid control (predictable bandwidth, not latency tolerant)

--- Pai Chou (UC Irvine)

Sophisticated big data analytics is a dominant workload. The data deluge demands sophisticated data computation and intensive data communication. Computation will be characterized by sophisticated machine learning and data analytics kernels to extract information from data. Communication will be characterized by large, but infrequently accessed data footprints that follow a zipf distribution. These workloads will be large, parallel/distributed, and diverse as users assemble a mix of libraries and frameworks to tailor solutions to domain-specific analyses.

--- Benjamin C. Lee (Duke Univ)

Nokia is focused on mobile devices. Significant computational workloads will include gaming, media streaming, augmented reality, and computational photography. Almost all workloads will have some network traffic, dominated by streaming video. Increasing numbers of always-on applications will increase network signaling activity (e.g. keep alive signals).

--- Per Ljung (Nokia)
In future years, applications related to health care, social media, gaming and entertainment, as well as data analytics will likely dominate. The network traffic patterns for this kind of applications exhibit high variability and potentially pronounced non-stationary signatures so performance analysis and optimization algorithms will need to incorporate time explicitly and exploit the various characteristics of the traffic patterns for accuracy and robustness reasons.  
--- Radu Marculescu (CMU)

Virtual worlds, games, business analytics and smarter planet applications will produce workloads that require very short response time. The main computation workloads will include graphics and ray tracing for visualization, physics modeling for world simulation, and data stream processing for smarter planet and analytics.  
--- Jose Moreira (IBM Research)

I expect a range of applications from dense matrix streaming workloads with regular predictable memory access patterns to sparse/graph applications with irregular unpredictable memory access patterns. Many applications will be characterized by a combination of signal processing and machine learning examples are visual recognition and natural language understanding.  
--- Kunle Olukotun (Stanford)

I believe we will have five broad application classes: enterprise applications (e.g., supply chain, ERP), web services (e-commerce, video streaming), data analytics/search (data warehousing, search), presentation/multi-media/games, and High-performance computing (life sciences, defense, simulations).
All these categories are going to be influenced by a common secular trend ? the explosion in data. Beyond increased volumes of data, we are likely to have to deal with diversity in data types (and associated computations), the need for much faster response times, and the emergence of more sophisticated data processing and analytics. In terms of impact on system design, this is likely to translate to need for increased data bandwidth ? at the memory, network, and computational levels; sometimes in combination.  
--- Parthasarathy Ranganathan (HP Labs)

Application Classes: I would broadly classify future workloads into two categories: one category enables ambient (or ubiquitous) computing through pervasive mobile devices, and the other that seeks to extract value from large amounts of data on the server-end. The first category will consist of workloads that closely mimic or build upon today’s web browser infrastructure. We are already seeing the emergence of such new platforms in the market with the release of HTML5, and other similar language paradigms and language extensions. The second category will comprise mainly of applications that rely heavily on recognition,
mining and synthesis (RMS) kernels (or analytical engines), which will require the ability to rapidly extract useful information and valuable trends from large amounts of raw data.

Dominant Workloads: Web search is a first class Internet-scale workload with a majority of users now using search engines to locate a website that provides them with their desired information. In order to support massive current requests, Google is estimated to operate over 1 million servers, which is approximately 2 percent of the world’s servers. But Web 2.0 workloads, such as social media, are quickly surpassing the demands of web search. Social media workloads, such as Facebook and YouTube, differ from web search in that they are more interactive and dynamic, creating a much more pliable domain for deploying and sharing various types of media. For example, YouTube users upload over 20 hours of video in under a minute. At present, Facebook and Google alone account for 14 percent of all US Internet traffic. At the other extreme of systems, most embedded computers run managed runtime environments, which is typically Java. There are 3 billion mobile phones that run Java. Also nearly all of the Blu-ray players run Java. Web cams, games, set-top boxes, lottery terminals, medical devices, parking payment stations, security checkpoint devices, etc run Java. Therefore, I believe management runtime environments will evolve and play an ever-increasing significant role in mobile systems.

--- Vijay Janapa Reddi (UT-Austin)

Platform agnostic Web apps based on HTML5 with lots of rich and dynamic media content.

--- John Shen (Nokia)

In the mobile space at least, the dominant workload is pretty clear: It's surfing the web followed by consuming media. There are a couple other apps that people use a fair bit, but these are the ones that really drain battery life. In data centers and on the desktop, I think it's much less clear what the “dominant” workload will be.

--- Steve Swanson (UCSD)

Mobile platforms will be #1 by volume. Top application on smart phones/tablets will continue to be the web browser for foreseeable future, though I expect gaming to continue to gain ground. The most exciting new applications in this space are in augmented reality/vision, which will have massive computation and memory requirements. Speech recognition will also grow; as most speech recognition algorithms explore large Markov models, they tend to be memory latency bound with 10's of megabytes of footprint. For all these mobile apps, a key question is the application partitioning between the client and the cloud.

In the data center, interactive web based applications and scientific computing will both continue to be important. Peak power is a key constraint for big science; energy management tends to be less of a problem. For interactive web apps, low latency will remain far more important than power/energy. Key pieces of infrastructure, like networking stacks, memcache, transaction processing databases will continue to be important.

--- Tom Wenisch (Univ of Michigan)
2. What are the applications/workloads that are likely to place special requirements on the CPOM (including web servers, high-performance computing, multi-player game playing, and embedded computing)?

"cyber-physical" systems that control everyday infrastructure and utilities (e.g., power line, water, gas, ..) will require ultra-reliable, latency-guaranteed network requirements and robust cloud support with redundancy. There will be another class where the local smart terminal (tablet, PC, etc) will manipulate a physical model (of an object under design, of a 3D model of Google map, etc) that will require fetching data subset on demand over the network from a giant data set in the cloud.

--- Pai Chou (UC Irvine)

Coordinate processor-memory design for big data applications. Big data analytics workloads, such as online search and distributed memory caching, pose challenges for future web servers. Future server architectures must coordinate application quality-of-service with architectural energy-efficiency. Similarly, scientific workloads pose challenges for future high-performance computing platforms. Since gains in memory bandwidth will continue to lag those in compute bandwidth, future supercomputing architectures must coordinate processor and memory system design. Moreover, supercomputing applications will face new challenges to accommodate shrinking memory bandwidth/capacity per processor core while avoiding the expensive communication of distributed memory systems.

--- Benjamin C. Lee (Duke Univ)

Telepresence
Augmented Reality
Virtual Assistants (beyond Apple Siri)

--- Mark Hill (U. Wisc-Madison)

High-level, portable OpenCL and WebCL code are currently not tuned for specific platforms, and may provide fertile areas for CPOM.

--- Per Ljung (Nokia)

The availability and possibility to gather and analyze huge amount of heterogeneous data, together with data mining algorithms will generate new workloads where the focus will shift from traditional centralized computing to distributed search, mining and actuation.
Health care type of application and computational molecular biology will pose new requirements since accuracy and speed of large scale distributed simulations at both microscopic and macroscopic levels will be needed to battle with rapid virus and diseases evolution. Web analytics and query together with click streams will require architectures to perform fast searching and online transaction processing. In addition, there will be a rich set of application seeking to determine near-optimal personalized recommendations, to detect patterns and trends for anomaly or abnormal behavior prediction, or to ensure a high degree of knowledge extraction and data mining from increasing heterogeneous data flow. Consequently, the distributed nature of these applications and the intrinsic optimization needs will require architects to enhance computing platforms with smart dynamic and decentralized optimization strategies for efficiency, robustness, and energy minimization.

--- Radu Marculescu (CMU)

All the applications above will place requirements on the CPOM. The objective will be to maximize the amount of computing performed per amount of energy, under the strict requirements of short response time.

--- Jose Moreira (IBM Research)

Data analytics workloads that require processing of huge graphs. Examples are social network analysis, bioinformatics, knowledge discovery

--- Kunle Olukotun (Stanford)

In my opinion, cross-layer power management will be key in *every* domain to get the next major multiplier improvement in power management. (See “Saving the world together, one server at a time,” Ranganathan and Chang, IEEE Computer, May 2011.)

--- Parthasarathy Ranganathan (HP Labs)

Enabling cloud computing applications, such as Web Search and Web 2.0 applications, comes at the severe cost of a data center energy crisis because of their large scale in number of nodes. The IT industry already creates about two percent of the world’s carbon emissions, and datacenters are noted as the fastest growing contributor of that footprint. The world’s datacenter carbon emissions are approaching those of countries such as Argentina and the Netherlands. Without effort to curb demand, the Environmental Protection Agency estimates that datacenter emissions will quadruple by 2020 to 340 metric megatons of CO2, which is just a few ranks below the top 10 countries’ total emissions. Thus, to sustain the growth of IT infrastructure, we must make CPOM advancements in the energy efficiency of our servers.

Nowadays, managed runtime environments, such as Java and .NET, are getting ignored by the new phones such as iPhone and Android primarily because of lack of performance and high
overhead, and consequentially low power efficiency. I doubt anyone denies the benefits of Java, but Steve Jobs has often been quoted as saying that “Java is this big heavyweight ball and chain.” If the performance of managed runtime environments can be improved, then more embedded devices are likely to use it. A part of our effort must involve understanding and solving problems associated with these managed virtual machines, especially embedded JVMs. Aside from phones, there are other embedded devices that run security applications, such as at checkpoints and security gates. Understanding and improving these workloads and platforms is crucial to computing that is becoming pervasive at a new scale.

--- Vijay Janapa Reddi (UT-Austin)

Same as 1. Other than computation, wireless communication will be a serious problem, especially as the industry move to 4G and always connected usage mode.

--- John Shen (Nokia)

I would divide the the class of "special requirements" into two classes: Static and dynamic. In the static case, fixed or slow-changing applications (e.g., those for which substantial changes in the application occur less frequently that users purchase new hardware), require us to tune vertically across the system layers to minimize power/energy consumption. The key issues here revolve around hardware specialization and hardware/software co-design. In the dynamic case, we have applications that vary quite quickly (i.e., on a time scale much shorter than the hardware life cycle). In these cases, the opportunities for power management through specialization are smaller, and designers will need to rely on software techniques to manage power consumption. It is likely that single systems will contain application components in both classes (e.g., a cell phone will run core software that changes infrequently, but the user may download many apps that may be updated more frequently).

--- Steve Swanson (UCSD)

See above. Mobile platforms need to have a 1-day battery life under arbitrary usage, hence, energy management is key. They are also thermal limited due to passive convective cooling. Standby power management is a mostly solved problem in mobile; what is more interesting is to conserve energy when devices are active or only slightly active (maintaining a link to the cell network is an example of "slightly active"). The constraints in the data center are all economic; there is no particular reason to impose a peak power cap on a server other than that it gets more expensive if the power budget gets really high. Improving the inter-rack bandwidth within the data center is a key challenge to make it easier to scale up applications. While networking power is not significant now, it is growing as a fraction of the data center power budget.

--- Tom Wenisch (Univ of Michigan)
3. What are the dominant application drivers and design platforms of CPOM (CMPs used for enterprise computing and data mining vs SoC’s used in PDAs and laptops)?

The dominant application (meta-)drivers will be those that
1. further blend the virtual/cyber world with the physical/analog world through control and actuation,
2. keeping distributed systems data-synchronized
3. high-definition media streaming or manipulation, including internet TV, high-def video conferencing
--- **Pai Chou (UC Irvine)**

Rethink platform design and scope for big data analytics. Conventional enterprise computing, which are typically memory- or I/O-bound, has benefited from more than a decade of efficiency research for CMPs. Many, small, efficient cores reduce Joules per operation while meeting throughput and latency targets. However, future data analytics workloads require more sophisticated computation not amenable to the performance trade-offs required from small-core efficiency. Achieving energy efficiency for these future applications will require looking beyond chip-multiprocessors to coordinated processor, memory, and network systems.
--- **Benjamin C. Lee (Duke Univ)**

Greatest growth will be in mobile devices and for high-bandwidth "natural" user-interface devices (e.g., an active wall). This growth will be complemented by infrastructure growth in the cloud.
--- **Mark Hill (U. Wisc-Madison)**

Mobiles are moving towards single chip implementations (including multiple radios) to lower cost and reduce area. Such platforms may also be used in tablets and laptops.
--- **Per Ljung (Nokia)**

More work will be generated in the back-end (data center) because of applications running on PDAs and laptops. The demands on the CPOM will be harder to satisfy in the data center.
--- **Jose Moreira (IBM Research)**
The enterprise platforms will focus on general purpose CMPs coupled with data parallel accelerators (GPUS (floating point) and streaming memory (data access)). The key limit to the performance of enterprise applications will be memory bandwidth. The mobile platforms will use scaled down versions of the enterprise platforms (modulo ISA) augmented with many special purpose functional units. The key limit to the mobile/embedded platforms will be power.  
--- Kunle Olukotun (Stanford)

Again, I think the volume economics of the industry will lead to a convergence of architectural approaches for both these markets towards SoCs across the spectrum. (The recent emergence of “microblades” with low-power mobile processors is an example of this trend.)  
--- Parthasarathy Ranganathan (HP Labs)

Application Drivers: To sustain the rapid growth of our digital global economy, IT services are scaling out their compute clusters or data centers with cheap commodity hardware based on processor technology. By exploiting declining prices and the increasing performance of commodity systems, infrastructure providers hope to improve their return on investment (ROI) through horizontal scaling (scale-out) of compute resources. And the smartphone in our hands nowadays is nothing but a pocket computer, which happens to also have the ability to make a phone call and to perform basic phone functions. The core of a modern smartphone (Tegra) is a powerful CMP processor that runs a complete operating system. This transformation in phone devices has enabled a variety of new and versatile applications. Platforms: Advances in chip multi-processors comprised of small cores provide opportunities for power efficiency. These smaller cores deliver throughput with better power efficiency when compared to their low-latency, high-performance counterparts. Recent advances in mobile computing provide small cores with even greater power efficiency. Other technologies, such as photonics, nonvolatile (NV) memory, 3D die stacking, multicore and heterogeneous architectures offer compelling new opportunities. These trends’ confluence encourages a fundamental rethinking of system design and operation for CPOM.  
--- Vijay Janapa Reddi (UT-Austin)

Mobile handsets will be the dominant driver due to the form factor and power constraints.  
--- John Shen (Nokia)

This is difficult to say, because it's not clear what "dominant" means. I would lean toward the mobile space, though, because for cell phones and tablets, power really is the key constraint. In the datacenter, it's important, but ultimately I feel like it's a close second to performance (whether it's latency or bandwidth). However, I think it's pretty clear that both the mobile and enterprises spaces have CPOM-related issues that warrant large scale research efforts.  
--- Steve Swanson (UCSD)
SoC's will continue to dominate mobile. I expect a moderate increase in the number of general purpose cores (we've only hit dual-core so far in shipping devices; 4 or 8 cores seems likely to happen). I expect a massive proliferation in special-purpose cores/IP, as this is one of the best ways to improve energy efficiency and performance simultaneously.

The data center market may evolve to offer two kinds of systems - Xeon style "beefy-core" servers with moderate numbers of cores and excellent single-thread performance and massively-multicore "wimpy-core" systems, such as the recent announcement from Calxeda/HP or the chip from SeaMicro. Despite predictions in many articles, beefy core is not going away, there are plenty of applications where wimpy cores cannot meet latency requirements. Whether the wimpy-core server market ever takes off is unclear, but it is promising as an approach for power/energy management.
--- Tom Wenisch (Univ of Michigan)

4. How do these applications affect the on-chip data traffic and stress on-chip resources?

The on-chip data traffic will increase by the applications but I think having a local energy buffer will reduce the stress on the supply lines because they will have lower peak power.
--- Pai Chou (UC Irvine)

Emphasize application accelerators and local communication for efficiency. Future data analytics applications may be amenable to accelerators that target commonly recurring software kernels. At present, processor-accelerator communication relies on expensive direct memory accesses (DMAs). In contrast, a more efficient cache-coherent shared memory may alter on-chip data traffic patterns and require new communication architectures. Stressing on-chip resources is likely more efficient than communicating via main memory.
--- Benjamin C. Lee (Duke Univ)

Multicast video (e.g., telepresence) will stress bandwidth, while augmented reality require low latency.
--- Mark Hill (U. Wisc-Madison)

Graphics intensive applications (i.e. games, video) dominate on-chip traffic and resources. Multiple loaded (but not running) applications implies significant data movement and cache fills when switching foreground apps. Multiple always-on applications similarly result in significant data movement.
--- Per Ljung (Nokia)

We foresee several specific characteristics of these applications running on large scale computing platforms: heterogeneity in utilizing computing and communication resources, fractality and non-stationarity coming from the real-time and interactive nature of these applications. Consequently, dynamic optimizations dealing with high workload variability are essential for future computing platforms.
--- Radu Marculescu (CMU)

Data will have to be processed on the fly against large models and databases residing in main memory. Ability to simultaneously stream I/O data and memory data will be a requirement.
--- Jose Moreira (IBM Research)

I don’t think on-chip data traffic is a problem
--- Kunle Olukotun (Stanford)

I think it is simplistic to assume that there is one single answer. There will continue to be a diversity of requirements specific to individual workloads. E.g., collaborative filtering (ala the Netflix grand challenge) will have significantly higher compute requirements relative to data bandwidth versus media streaming which will have significantly higher bandwidth requirements compared to compute. (See “Data Dwarfs: motivating a coverage set for future large data center workloads.”)
--- Parthasarathy Ranganathan (HP Labs)

Continuing to use commodity hardware to support emerging applications comes at the high cost of both energy consumption and increased unreliability. Commodity hardware is not designed under strict guidelines to attain the highest possible energy-efficiency and reliability. Instead, manufacturers relax operational standards to mitigate cost overheads. This results in persistent energy inefficiencies during normal operation. Moreover, current hardware trends (diminishing feature sizes, lower voltages, and so forth) are increasing the susceptibility of devices to transient bit flip errors, and thermal and voltage fluctuations. Even power management cycling strongly affects component lifetimes due to thermal and mechanical stresses. These forces combined together with increasing cloud component count imply higher inefficiencies and error rates will be likely in the future.
--- Vijay Janapa Reddi (UT-Austin)
Consume too much energy.
--- John Shen (Nokia)

On-chip interconnects in SoC will become more sophisticated - AMBA is a terrible way to connect IPs. I predict that on-chip crossbars and higher-radix interconnects will replace meshes, rings, and other simpler NoCs, because they provide lower latency and less latency variance, generally require less total buffering, and can be easily extended to prioritize traffic/ reserve bandwidth.
--- Tom Wenisch (Univ of Michigan)

5. How to characterize them from computer architectural and programming model perspectives?

The architecture may need to provide mechanisms for charging control and remaining energy sensing from software. It may need to generate an interrupt when the remaining locally buffered energy drops below a threshold. The programming model can either map the peak power constraint and energy quota to time or instruction count or similar mechanisms.
--- Pai Chou (UC Irvine)

Infer hardware-software interactions via synthetic benchmarking. Robust synthetic benchmark generators can anticipate behavior in and between processors, memory, and network architectures. By generating synthetic activity, designers can illuminate the full spectrum of hardware-software interactions. With sparse measurements, statistical inference can infer hardware-software interactions to support coordinated design. With inferred interactions, a variety hardware system architectures and/or software programming models can be fully characterized. For example, this approach makes tractable auto-tuning software on reconfigurable hardware.
--- Benjamin C. Lee (Duke Univ)

Not sure how to answer.
--- Mark Hill (U. Wisc-Madison)

OpenCL and WebCL are newer programming models designed to provide high performance to multiple platforms. Auto-tuning of given OpenCL code for a particular platform is of considerable interest.
Performance analysis models need to account for workload heterogeneity, fractality and non-stationarity. Such a performance analysis framework can be build on statistical physics concepts such as master equations which later can lead to approaches for efficient dynamic optimization algorithms.
--- Radu Marculescu (CMU)

Streaming programming models are a good start, but there needs to be more. Large memory will play an important role.
--- Jose Moreira (IBM Research)

Ideally, programming models should be dictated by the computational domain using domain specific languages. The details of the architecture should not be visible to the programmers.
--- Kunle Olukotun (Stanford)

I think this is going to be one of the most important challenges. We will need to revisit traditional assumptions around balance, but we will also have to start thinking of new metrics and models to reason about what we do with computers.
--- Parthasarathy Ranganathan (HP Labs)

Many emerging cloud computing workloads are proprietary. Embedded workloads in many Web 2.0 devices are also proprietary and/or classified. Developers of these applications typically do not (or cannot) to share their applications as benchmarks during hardware developmental stages. However, at the same time, they desire hardware developers of their platforms to design and optimize hardware whose capabilities are matching to the requirements of their workloads. Therefore, it would be desirable to have workloads that are representative of emerging proprietary applications available during hardware development. Availability of synthetic or proxy workload that represents performance, power, reliability characteristics of the original workload will help hardware software co-design and co-optimization for CPOM.
--- Vijay Janapa Reddi (UT-Austin)

Nothing like traditional approach. Whole new paradigm is needed.
The two places where these issues are most important (mobile and the enterprise) both pose enormous problems from a characterization and modeling perspective. For mobile devices, the platforms are relatively closed and are not amenable to simple instrumentation. Likewise, the software stacks they run are very complex. Existing profiling and simulation frameworks that architects have relied on for modeling are really not up to the task. For instance, on Android phones, a single "app" comprises a virtual machine, a JIT, a bunch of libraries, and a collection of hardware accelerators. Collecting an accurate picture of how all these pieces are interacting is very hard. Persuading industry to provide access to open prototyping platforms would go a long way toward solving these problems.

In the enterprise the problem is even more challenging. There, the applications are more complex and, in addition, they span hundreds or 1000s of nodes. This multiplies the number of components that need monitoring, modeling, and management if we are to create a comprehensive approach to CPOM. Existing cloud research infrastructure may be of use here, but simulation is in tractable.

--- Steve Swanson (UCSD)

Technology:

1. What research issues must be addressed in this area? How do we achieve power or energy efficiency under latency, bandwidth, thermal, or cost?

Service migration, latency optimization, hybrid energy sources for optical/mechanical subsystems, energy storage, energy harvesting, energy conversion, power (electrical or other) transmission

--- Pai Chou (UC Irvine)

Understand and navigate dependencies between emerging technologies. As technologies evolve in parallel, system architects must identify dependent assumption, define design spaces, and propose architectures robust to shifting assumptions but yet improve performance and efficiency. For example, emerging memory technologies promise qualitative improvements in several performance metrics but the sensitivity to technology assumptions is often not clear. For example, advances in memory cell technology must be coordinated with those in integration and packaging.

--- Benjamin C. Lee (Duke Univ)
Can we exploit energy-efficiency of specialization in compelling, flexible products?
--- Mark Hill (U. Wisc-Madison)

The primary resource constraint is energy. There can be a run-time tradeoff between QoS and energy by balancing computation and communication. For example slower processors with higher energy efficiency can be used, similarly slower radios with lower range can be used. High local computational cost can be traded for cloud communication costs.
--- Per Ljung (Nokia)

Relying on accurate performance metrics that take into account workload characteristics and capture architectural constraints it is essential for defining dynamic optimization methodologies. This tight coupling can lead to optimal solutions. We would need though specialized efficient low power cores to help solving such dynamic optimizations.
--- Radu Marculescu (CMU)

We need to be able to dynamically adapt between low latency and high throughput computing. Not clear this can be done with the same design.
--- Jose Moreira (IBM Research)

Need to develop mechanisms to extract and exploit domain knowledge in the application to optimize performance per watt. Need to use domain knowledge to drive the design of specialized architectures.
--- Kunle Olukotun (Stanford)

Three major likely recurring themes: leverage energy-efficient technologies; revisit systems architecture grounds-up for balanced design optimized for sweet spots of energy efficiency; revisit hardware-software separation of functionality, in particular focusing on energy-inefficient software layers and interfaces.
--- Parthasarathy Ranganathan (HP Labs)

Power dissipation in massively multicore systems will become a key limiter to selecting processor designs. Already, today’s relatively small multicore systems are reeling with high thermal density and power dissipation issues which negatively impact performance, cooling
costs, and chip lifetime. Tomorrow’s large multicore systems with hundreds of cores will have even higher thermal gradient, thermal density and dissipation overhead. The problem becomes especially challenging when we realize that techniques used to lower power dissipation such as voltage scaling decrease noise margins and end up increasing susceptibility to faults. Moreover, massively multicore systems in the coming years will be plagued by a high 10-30% device failure rate. Under the duress of high error rates, current schemes and research aimed primarily at systems with low error rates may not be useful. Novel schemes will be required that can tolerate high failure rates and still provide useful chip operation.

The challenge for us is to transparently eliminate inefficiencies found in commodity hardware that are tolerated in order to reduce price or, in general, to optimize for criteria other than performance. We must design future systems such that they recognize and respond to interaction between an application’s dynamic activity, the underlying platform, and the behavior of other system software. The system must analyze the causes of fluctuating power demands in multicore and heterogeneous systems, satisfy reliability constraints, and apply the resulting analysis to a dynamic framework for optimization. The eventual result is a feedback-directed optimization system that balances power and performance, and other constraints across the entire platform at the microsecond- and nanosecond-level.

--- Vijay Janapa Reddi (UT-Austin)

Need to take a broad systems approach to reduce both active power and standby power.
--- John Shen (Nokia)

At the chip level, we need more aggressive power management components than are currently available. These include more efficient and faster power-gating solution, faster mechanism for changing processor voltage and frequency, and finer-grain monitoring of on-chip energy and power consumption. Since application behavior varies on a fine grain in many cases, the CPOM mechanisms we develop must operate at similar timescales. Otherwise, much of the potential benefits of CPOM systems will go unrealized.
--- Steve Swanson (UCSD)

2. What level and amount of re-configurability (both at the circuit and architecture levels) are needed to enable power-efficient operation of a target system under various workload conditions and reliability/fault tolerance requirements? What is the support needed at the hardware and system software level to enable such reconfiguration?

The architecture level should ultimately support drastic mode change by switching to alternative sub-architectures (e.g., 16-bit, 32-bit, 64-bit, GPU) under program control. Check pointing and background syncing are needed in order to achieve fault tolerance. Local, possibly on-chip or
on-board, energy buffering will be another very important feature for reliability while improving power-efficient operation. I believe integrated circuitry that handles dc-dc conversion, charging, (local) energy storage, and bus power transfer will be needed. Software schedules will need to take peak power into consideration and manage power buffers across multiple power domains to help address the peak load.

--- Pai Chou (UC Irvine)

Define hardware-software abstractions for coarse-grained reconfigurability. Reconfigurability should be application-driven and coarse-grained. As energy efficiency is enhanced with specialization, preferred accelerators will vary across applications. By using reconfigurable fabrics for accelerators, some measure of generality might be possible. To implement control mechanisms, interactions between system software and hardware mechanisms must be well understood. Designed separately, these software and hardware mechanisms may produce unintended consequences or sub-optimal control. Rather, clean abstractions are needed. For example, hardware is responsible acts and monitors while software decides.

--- Benjamin C. Lee (Duke Univ)

Higher energy efficiency usually implies specialization. Reconfiguration usually implies broader applicability, and therefore worse energy efficiency. Faulty resources in network-on-chip or many-cores may need reconfiguration Software defined radio with configurable encoding and modulation subsystems are also relevant. It is unclear if HW fault detection is required. Programming different routing tables should allow for circumventing non-operational links or cpus (with performance penalty).

--- Per Ljung (Nokia)

First and foremost, we need to enhance computing platforms with various cheap sensors that can provide dynamic information about the state of the system. All this information can then be fused by several decentralized decision cores to determine the right level of re-configurability.

--- Radu Marculescu (CMU)

Fault tolerance and reliability must be selective: pay for what you use. Reconfigurability between low latency and high throughput will be a must.

--- Jose Moreira (IBM Research)
I think specialized architectures with limited reconfigurability/programability are required for
dramatic improvements in power efficiency. New programming models and compilation
technology is required to hide the resulting heterogeneity from the programmers.
--- Kunle Olukotun (Stanford)

I would broaden this discussion to include systems architecture and software as well. I think
there is a fundamental tension between the general-purpose and volume benefits of
reconfigurable designs and the energy efficiency benefits from specialized accelerator-based
designs. The key challenge will be identifying a set of base primitives that are general enough
yet efficient enough. Doing this at every level is hard enough, but doing this as a global
optimization function across layers will be even more important. In terms of support needed for
reconfiguration, I see three main areas: rich measurement and monitoring infrastructure;
accurate models for resource prediction and prescriptive feedback; control algorithms and
policies to manage the actuators in aid of reaching a constraint-driven objective function.
--- Parthasarathy Ranganathan (HP Labs)

The keys for energy-efficient design fall into two categories: (1) reducing waste, which is
a pure gain, and (2) carefully balancing the tradeoff between performance, power and
reliability. To effectively enable this, future systems will require both hardware and software
support. At the circuits layer we require sensors to gather pertinent information. At the
architecture layer we must provide fast and reactive mechanisms to prevent immediate
catastrophic failures. At the software layer we must leverage its global knowledge to
make smoother, and more persistent and cost-effective, transformations to execution that
improve hardware operation. The hardware must provide hooks that allow such software-
directed control of hardware execution. Across all these layers, there must be a synergistic
approach to adaptive power management.
--- Vijay Janapa Reddi (UT-Austin)

Must leverage heterogeneity in communication and computation.
--- John Shen (Nokia)

Determining the correct about reconfigurability to minimize power and/or energy over very
large workloads is an especially challenging engineering task. While specialization (which
reconfigurablity enables to some extent) can save a lot of power, the reconfiguration hardware
itself is pure overhead The approach we have taken is to identify classes of applications that can
all be served by hardware with moderate reconfigurability. It's pretty clear that adding much
more than this hurts energy and EDP. On the other hand, reducing reconfigurability reduces
application coverage. Finding the right balance point requires sophisticated, scalable program
analysis and compiler techniques to automatically identify program classes and determine the most efficient forms of reconfigurability to provide.

--- Steve Swanson (UCSD)

Determining the correct amount of reconfigurability to minimize power and/or energy over very large workloads is an especially challenging engineering task. While specialization (which reconfigurability enables to some extent) can save a lot of power, the reconfiguration hardware itself is pure overhead. The approach we have taken is to identify classes of applications that can all be served by hardware with moderate reconfigurability. It's pretty clear that adding much more than this hurts energy and EDP. On the other hand, reducing reconfigurability reduces application coverage. Finding the right balance point requires sophisticated, scalable program analysis and compiler techniques to automatically identify program classes and determine the most efficient forms of reconfigurability to provide.

--- Tom Wenisch (Univ of Michigan)

3. What CAD tools are needed to realize CPOM and to design systems utilizing CPOM?

CAD tools will need to be able to model discrete power switching (such as PowerSim) for modeling conversion while at the same time use the workload to drive the underlying system configuration.

--- Pai Chou (UC Irvine)

Parameterize hardware descriptions to quickly define and refine designs. Parameterized design frameworks are required. As specialization and heterogeneity become increasingly popular, design variants must be easily defined and evaluated. Rather than supporting the creation of a design instance, CAD should support the creation of design families. Analogous to software code generators, which specify a baseline implementation and automatically generate code for optimized variants, hardware chip generators should be able to automatically generate optimized hardware descriptions. Such hardware generators leverage existing hardware description languages with new scripts and optimizers that define and refine designs.

--- Benjamin C. Lee (Duke Univ)

Mobile systems comprise low-level hw resources, firmware, and operating system (e.g., communication stacks, schedulers, resource management, UI). Each of these is typically developed by independent vendors making cross-layer optimization difficult. CAD tools that could generate appropriate APIs for each subsystem would be beneficial.

--- Per Ljung (Nokia)
CAD tools are likely to change and try to capture the dynamic component of applications in order to provide a higher level of savings. CAD tools need to offer the possibility to optimize for re-configurability, for dynamic control of resources or to ensure robustness and fault-tolerance.
--- Radu Marculescu (CMU)

CAD tools for 3D designs need to evolve and get much more versatile and robust.
--- Jose Moreira (IBM Research)

We need CAD tools that can make a calculated trade-off between power, performance and reliability based on real-world operating conditions, rather than assuming worst-case peak operating points for system design. For instance, we may want a total-sensitivity based standard cell DFM flow. Sensitivity metrics may include device/transistor criticality, non-rectangular gate impact due to lithography printability, process-variations (e.g., dosage and defocus), and so forth. Such total sensitivity modeling built into a cell layout optimization engine can help us minimize the performance gap between peak operating conditions, and normal behavior.
--- Vijay Janapa Reddi (UT-Austin)

CAD tools that can support automatic construction of the kinds of hardware described above do not currently exist. If reconfigurability and specialization are to be widespread we need much more sophisticated and automated support for creating and verifying efficient hardware. In effect, if we are going to realize ubiquitous specialization for CPOM, building the hardware will need to be almost as easy as compiling software.
--- Steve Swanson (UCSD)

3D stacked chips will happen (for logic as well as memory) eventually. CAD tools need to evolve to keep up.
--- Tom Wenisch (Univ of Michigan)
4. How is one to achieve optimizations that combine dynamic control, algorithmic transformations, and compiler transformations? What are the research issues in this area?

The architecture needs to provide the mechanisms for dynamic control and interrupts for resource changes (power, energy). The compiler will need to determine the possible peak-power and energy budget (just like compilers determine the register or memory footprint, or instruction count estimation). At job level, the OS at runtime can determine when to context switch. Interrupts can be generated by hardware to notify software on energy changes. Research issues include holistic, integrated modeling of (dc-dc) converters, chargers, storage along with load characterization in the context of local power transmission among subsystems. Associated with the new platform are the power management schemes involving joint scheduling of data communication and energy buffering. The research first applies to local wired LANs (with bus power) and will be extended to systems-on-board and systems-on-chip.
--- Pai Chou (UC Irvine)

Infer hardware-software interactions via hardware-software generators. Coordinated optimization requires an understanding of hardware-software interactions. The performance and efficiency effects of a new algorithm or compiler transformation depend on hardware activity. Given sufficiently diverse measurements of software behavior, hardware-software interactions can be statistically inferred. To observe such diversity, research is needed in automated hardware-software generators. With generated measurements, research in inference for composable models is needed.
--- Benjamin C. Lee (Duke Univ)

Dynamic, just-in-time optimizations or auto-tuning are very relevant. They allow a single high level source (e.g. html5, opencl) to be used by several dissimilar platforms. There has been significant effort (e.g. UCB Parlab seijits) on high performance computing (e.g. auto-tuned 2x higher throughput than FFTW) but very little activity for mobile platforms. There has been little activity on energy driven by computational load, whereby the potential of local power islands can be varied from sub-, near-, and super-threshold dependent on the rate of IO events.
--- Per Ljung (Nokia)

A crucial aspect is to have a simplified, scalable yet secure operating system that allows access and optimization at both ends: application and architecture. Application partitioning, clustering, mapping and scheduling cannot be done independently of platform availability and vice versa. The entire network stack protocol needs active research.
--- Radu Marculescu (CMU)
Compiler optimization and algorithmic transformations need to be more measurement and feedback driven. That matches well with dynamic control.

--- Jose Moreira (IBM Research)

Need domain specific compilers that can perform algorithmic transformations and pass important information to the runtime systems. The runtime system should use compile time and runtime information to adapt the application execution to optimize performance per watt.

--- Kunle Olukotun (Stanford)

I think one of the biggest research issues, in my opinion, is the need for a cross-layer power management architecture that allows different optimizations to be integrated into a global design (See “No Power Struggles: Coordinated multi-level power management for the data center,” ASPLOS 2008)

--- Parthasarathy Ranganathan (HP Labs)

I envision an ultimate runtime management framework that co-exists with system software, but which continuously monitors and dynamically tailors executing program threads’ code to meet application- and system-level power, performance and reliability constraints. It will adaptively track, map and move data to computational resources (or vice versa) to maximize energy efficiency. To enable such a system we must identify the type of hardware support required, the type of information exchange between the hardware and software layers, and the optimizations themselves.

--- Vijay Janapa Reddi (UT-Austin)

I touched on compiler issues above. I'll leave the algorithmic and control theory bits to folks with more experience there.

--- Steve Swanson (UCSD)

The more the hardware can do without any involvement from software or compiler, the better. The beauty of clock and power gating in Nehalem is that it just works - the cores’ power draw goes way down whenever you hit a HALT instruction in the idle loop without any software involvement.

--- Tom Wenisch (Univ of Michigan)
5. How the power efficiency of a particular system scales for a given class of application workloads once the overall system performance has crossed a threshold.

Latency (in computation) could be achieved by speculative parallel execution and anulling wrong results; or by hashing cached results instead of computation in a large address space. Interactive systems need to exploit perception by prioritizing what the user perceives (especially on wakeup or limited bandwidth conditions) rather than fair scheduling. Compute-bound systems need to schedule, allocate, migrate tasks.

Actually, one unexploited technique is local energy buffering for smoothing out the power transmission. The optimal way to transmit power is to keep voltage high, current low, and power = V * I = constant. If supply power can be smoothed out by charging a local energy storage element instead, it could lower peak power.

--- **Pai Chou (UC Irvine)**

Quantify marginal costs to assess efficiency scalability. Performance and power trade-offs are best measured using a marginal cost approach, reflecting the percentage increase in energy for every percentage increase in performance. This approach highlights the rapidly increases costs of performance and identifies efficient strategies for achieving a performance target. For example, high-performance processors with extensive support for speculative execution (e.g., wide superscalar and deep pipelines) may incur higher marginal energy costs than simply increasing the supply voltage.

--- **Benjamin C. Lee (Duke Univ)**

We need new models of how systems work. See for example the HPCA 2012 paper on "Computation Sprinting" that designs for burst processing.

--- **Mark Hill (U. Wisc-Madison)**

CPU/GPU scales well and then saturates. Radio communication incurs large startup overheads, and large overheads for small packets. The cellular radio system power mode is determined by the network (not the handset) and carriers typically prioritize reducing network signaling rather than the handset power consumption.

--- **Per Ljung (Nokia)**
Relying on accurate performance models offers the possibility of performing sensitivity analysis and determining what solution sets are viable. It can also provide insights into how the performance scales as a function of either application complexity or architectural constraints.
--- Radu Marculescu (CMU)

That depends on the application. Some traditional web applications have reasonable linear scaling, whereas the computational demands for business analytics can grow faster than linear with the size of the input.
--- Jose Moreira (IBM Research)

Given that most systems don’t scale and power efficiency saturates after a certain threshold, I think the bigger question is how do we design systems so that power-efficiency scales well both up and down. I think the key to this is to design system with fungible modular building blocks designed for the sweet spots of energy efficiency that can be combined together as scalable ensembles to achieve the level of performance (reliability, etc) that the workload needs.
--- Parthasarathy Ranganathan (HP Labs)

When performance demands are satisfied, the system should automatically switch into power saving mode. Only minimum performance guarantees must be met, beyond which the system should exploit all available slack to reduce power/energy consumption; there is no obvious benefit to delivering performance beyond a user or application’s needs. Consider dynamic voltage and frequency scaling (DVFS) in a multicore chip. Future cores will likely have independent control of voltage and frequency, so the system must investigate the trade-offs between performing DVFS on individual cores or all at once. This may depend on whether the workload is homogeneous or heterogeneous. By understanding a system’s workloads, the dynamic runtime environment can schedule jobs to optimize power and performance.

Another example is cooperating vs. non-cooperating multi-threaded server programs. If a program is heavily multi-threaded, dynamically identifying and speeding up sequential portions of the code on a large brawny core could significantly enable scalable energy-efficient parallel execution while non-critical threads run on low-powered cores. Similarly, in a 3D stacked architecture that offers many layers of memory with varying performance and power levels, it may be possible to dynamically change which memory resource (DRAM or nonvolatile memory) a program should utilize according to its runtime behavior and resource constraints.
--- Vijay Janapa Reddi (UT-Austin)
6. Note that all innovations in architectures and micro architectures that reduce power must take into account potential impact on other critical aspects (e.g., latency, reliability, etc.). Where do you see trade-offs? Where do you see mutual benefit?

Exploit software optimizations that benefit both performance and efficiency. Software and application optimizations benefit both performance and efficiency by computing the same answer with less computation. Examples include software auto-tuners and approximate computing. Similarly, application-oriented specialization can reduce the overheads of instruction/data movement by effecting tens to hundreds of operations for each invocation of an accelerator. In contrast, performance-efficiency trade-offs arise when simply increasing the size of existing resources or gating un-used resources.

--- Benjamin C. Lee (Duke Univ)

Trading communication for computation is viable, for example to save energy by offloading, or to run complex games in the cloud that require more computational power than is available locally. The former case would normally see higher latency. A web proxy that provides compression and aggregation can reduce the total number of data packets sent, which can reduce a user's monthly data consumption and simultaneously reduce energy. Using subthreshold or other circuit techniques trades area for low power. Similarly using streaming processing can increase aggregate throughput, but results in larger startup/takedown latencies.

--- Per Ljung (Nokia)

In general, defining restricted optimization problems does not offer the best solution. Instead, using accurate performance models can help to define global optimization problems that take into account latency, reliability and power consumption. Only then the question of tradeoffs can be posed and properly solved given a set of available resources.

--- Radu Marculescu (CMU)

Tradeoffs and mutual benefit are possible if the requirements for latency, reliability, etc can be satisfied selectively. For example, only pay processor redundancy in critical sections of an application.

--- Jose Moreira (IBM Research)

I think architecture specialization can be used to improve performance and power efficiency at the same time. The key is managing the impact of specialization on the programming model. Domain specific languages and compilers provide the answer here.
--- Kunle Olukotun (Stanford)

I think the ultimate goal of the system design is “SLA/TCO”? where the service-level agreement defines the allowable tradeoffs across multiple dimensions at the performance, reliability levels and the total cost of ownership defines the tradeoffs across energy efficiency, design and operation complexity, manageability, etc.
--- Parthasarathy Ranganathan (HP Labs)

We must leverage application knowledge either at the binary level or the intermediate representation-level to dynamically make trade-offs between power and performance and other constraints. I believe that every opportunity will exhibit some level of slack or tolerance. We must identify and exploit these tolerance opportunities to minimize power consumption, or sustain performance without consuming additional power. I believe the dynamic system that I described above will be able to achieve this goal.
--- Vijay Janapa Reddi (UT-Austin)

Must maximize: \( \frac{\text{(user experience)}}{\text{(energy consumed)}} \)
--- John Shen (Nokia)

With respect to performance, our experience has been that performance cost of features that improve CPOM are minimal. The larger challenge, as I mentioned above, is to device microarchitectural support for CPOM that operates on the same scales as program behavior variation. In this case, it is not the micro architectural features so much that slow down performance, but the introduction of software that controls them (and their potential frequent use) can have a much larger impact.
--- Steve Swanson (UCSD)

Saving energy almost always costs performance, and performance and time to market will almost always trump energy efficiency (and, I would argue, even reliability) as a concern. Anecdotally, as soon as a data center operator receives a new server, the first thing they do is shut all power management features off to make sure that they don't cause performance anomalies.
--- Tom Wenisch (Univ of Michigan)