NSF XPS Workshop Questions

Jason Mars, Scott Mahlke

(a) Applications: What are the applications that motivate future systems? What advances in algorithms and programming systems will support these applications?

Data-intensive applications will consume the largest fraction of future compute cycles. These include all things machine learning, computer vision, audio/video processing, etc. We need to improve the hardware and software stack to enable more sophisticated approaches in as real-time as possible. Much of this data will not be stored or transmitted but rather analyzed on the fly, near the collection point.

(b) Systems: What hardware architectures and distributed systems will support future applications? What challenges do we face in design and management?

Embracing specialization will be key in the design of future systems for these applications. Compute substrates including, but not limited to, GPUs, manycore like Phi, FPGAs, and ASICs must be explored. Energy consumption is probably the largest challenge to process larger and larger volumes of data without increasing the energy footprint.

Programmability of these systems is a big issue. As hardware becomes more heterogeneous, general purpose software will not be possible any longer without the emergence of a new class of tools for mapping high level programs across heterogeneous hardware. Such hardware often requires distinct software structure, and organization, even source code language is often different for different devices (e.g., CUDA or OpenCL for GPUs). As hardware specialization becomes more diverse, separate language, compiler, and run-times for each class of device is not feasible. A unified software stack will be necessary to provide software developers a common platform to develop applications/libraries along with compiler tools to map and performance optimize code for the varying targets.

(c) Technologies: What emerging technologies will change fundamental assumptions in hardware and software? What constraints disappear? What challenges arise?

The emergence of correctness or quality of result as a new design parameter has the potential to reshape how we think about both hardware and software. Quality has always been a knob in signal processing, but this will proliferate to many more computing domains. “Good enough” will emerge as the necessary quality metric, such that the maximum gains can be realized by reducing output quality as far as possible, while still meeting some minimum threshold of meaningful output.

(d) Methodologies: How should we perform interdisciplinary research that spans applications, systems, and technologies? How should abstraction layers evolve?

We’ll need to collaborate across disciplines of course. However, it is important for Architects and system designers to embrace an approach where we gain expertise in the application domains themselves. For example, we should learn about ML techniques, computer vision, and NLP techniques as well.
(e) Risks: What are the risks that threaten the success of XPS research directions? How do we guard and hedge against these threats?

Papers will be published, students will be educated, and new results and advancements will occur, regardless of whether the research ideas are successful. Risks include design complexity – researchers tend to focus on new/novel ideas to get their papers published rather than cost, verification, correctness, etc. that are necessary in industry. Closer working with industry is one way to be more aware of risks, but also ensure that ideas can pass tough technical scrutiny and evolve when practical limitations are apparent.