Exploiting Parallelism and Scalability

Report on an NSF-Sponsored Workshop[1]
June 1-2, 2015, Arlington VA

Co-Chairs: Wu-chun Feng, Benjamin C. Lee
Steering Committee: Kunal Agrawal, Rastislav Bodik, Luis Ceze, Lingjia Tang

1 Executive Summary

Information technology is quickly approaching an inflection point at which previously reliable drivers of computational capability encounter fundamental challenges, and innovation across the spectrum of hardware and software will drive the future advances that society expects from computing. Although Moore's Law may produce another few generations of smaller transistors, the end of Dennard Scaling means that smaller transistors do not necessarily improve performance or energy efficiency. In this post-Dennard era, interdisciplinary research in algorithms, architectures, systems, and technologies is required to sustain computing's exponential trajectories in performance and efficiency.

In 2013, to address these challenges, the National Science Foundation launched a new research program -- Exploiting Parallelism and Scalability (XPS). XPS researchers from across the United States began interdisciplinary projects that integrated expertise from across the hardware stack. In 2015, XPS researchers convened in Arlington VA to discuss the past, present, and future of parallel and scalable computing. Participants discussed five broad questions that define fundamental challenges in computing.

1. **Applications.** What are the applications that motivate future systems? What advances in algorithms and programming systems will support these applications?

2. **Systems.** What hardware architectures and distributed systems will support future applications? What challenges do we face in design and management?

[1] This work is supported by NSF grant CCF-1451021. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsors.

3. **Technologies.** What emerging technologies will change fundamental assumptions in hardware and software? What constraints disappear? What challenges arise?

4. **Methodologies.** How should we perform interdisciplinary research that spans or co-designs applications, systems, and technologies? How should abstraction layers evolve?

5. **Risks.** What are the risks that threaten the success of XPS research directions? How do we guard and hedge against these threats?

The findings from the XPS workshop are summarized in this report. XPS researchers comprise a vibrant community that finds compelling reasons to coordinate innovation across hardware and software. Indeed, the case for coordination is stronger today than it was a few years ago. Many of the newest advances in computing have integrated previously disparate pursuits --- programming models for approximate architectures; architectures for emerging non-volatile memories; architectures for specific application domains; machine learning and economic game theory for systems management --- and opportunities abound for further research that re-thinks abstraction layers for performance, efficiency, and scalability. In the future, XPS research is needed to coordinate advances in five fundamental areas.

1. **Applications and Algorithms.** XPS research needs to address grand societal problems with new algorithms that provide the desired answers with less work and with programming systems that provide clean abstractions between elegant programming models and sophisticated run-time systems.

2. **Architecture and Systems.** Design for manageability anticipates management challenges during design. In architecture, research is needed in design and optimization to produce efficient and heterogeneous hardware components. In systems, research is needed in allocation and scheduling to ensure service quality for competitive users.

3. **Technologies.** Research is needed to exploit emerging technologies. This includes parameterized models to accommodate the inherent uncertainty in new technologies, especially those that might qualitatively change the way systems hold data (e.g., non-volatile memories and storage) and communicate data (e.g., packaging and interconnection networks).

4. **Methodologies.** XPS research must continue to be interdisciplinary, cutting across multiple abstraction layers in the hardware-software stack. Technology research should influence the design of balanced architectures, and architecture research should influence the design of balanced systems for next-generation applications.

5. **Risks.** As the landscape of computing changes, XPS research must maintain a broad perspective while avoiding pitfalls. Diminishing marginal returns and variants of Amdahl's Law require holistic research strategies. Communicating and building a community with other researchers, across academia, government labs, and industry, are prerequisites for success.

These findings highlight the breadth of research required to lay a foundation for computing in a post-Dennard, post-Moore era. Over the past few years, XPS researchers have collaborated to advance these priorities for innovation. Further sustained research in electrical engineering and computer science will build a strong foundation for future, yet-to-be-imagined, applications of computing.

2 Applications

Parallel and scalable computing lays a foundation for qualitatively new and significantly enhanced applications that change the way we interact with people, data, and the physical world. XPS research creates new solutions for societal problems, enables new software applications, and defines our approach to computation with new algorithms. The distinction between domains, applications, and algorithms helps define research scope and prioritize future research questions. For example, a domain might be drugs customized for individual patients; an application might be DNA sequencing and analysis; and an algorithm might be string similarity matching. Inspired and motivated by domains, applications, and algorithms, XPS research makes broad and lasting impacts on societal challenges.

Although future applications may evolve and emerge in unanticipated directions, successful XPS research distills key design objectives and computing frameworks that generalize beyond any specific application. XPS research in programming systems and software optimization is needed to adopt increasingly diverse hardware while meeting performance, power, and efficiency objectives. Capabilities developed for one class of applications will translate into insights and strategies for many others.

2.1 Domains

XPS research is needed to lay foundations for emerging and pervasive application domains that exploit big data analytics and an internet of things. First, an era of big data demands sophisticated analytics to translate bits into knowledge and insight. Big data analytics is broadly applicable, including revolutionizing the provision of health care, deepening our understanding of social networks, and accelerating our personal productivity. In health care, statistical machine learning on electronic health records will produce intelligent and personalized treatment, thereby reducing costs and improving patient outcomes. In social networks, graph analysis on interconnected populations will explain the diffusion of ideas. In personal productivity, digital personal assistants will combine big data with context, analyzing audio and video to provide knowledge precisely when it is needed.

Second, the internet of things demands pervasive and coordinated sensing that supplies large, diverse datasets for subsequent analysis. Sensing and computing are increasingly integrated and embedded into the physical world, providing qualitatively new capabilities in monitoring health and managing physical infrastructure (e.g., smart homes and cities). Individuals will carry an increasing number of ever more capable devices, which can be re-purposed or supplemented to monitor physiological parameters, track physical activity, and detect diseases early. Buildings and cities will increasingly rely on environmental sensors and actuators to automatically provision resources and utilities.

The representative domains we describe highlight only a small fraction of the societal challenges and opportunities that require fundamental advances in parallel and scalable computing. Fortunately, computer science is a science of principled abstractions, which means that an advance for one domain will supply extensible frameworks and trigger complementary advances in other domains. To realize this vision, foundational XPS research is needed in software and hardware, with tight coordination and flexible interfaces across the hardware-software boundary.

2.2 Applications

Software applications provide specific solutions for specific challenges within a societal problem domain. In the future, applications will be required to analyze massive datasets, perhaps streaming from distributed sensors, to infer broad and significant trends. XPS research is needed to exploit parallel and scalable computing and produce responsive and timely solutions. To illustrate these research challenges, we describe three classes of representative applications --- statistical machine learning, contextual

analysis, and scientific computing --- that will require foundational and coordinated advances in software and hardware.

Statistical machine learning infers relationships between data. XPS research is needed to translate parallelism and scalability into methods for efficient learning along several fronts. First, statistical approaches must scale on distributed systems while preserving statistical rigor. Research is needed to parallelize methods for classification, regression, and clustering while producing unbiased estimates for model parameters. New libraries (e.g., Spark) have improved scalability, but research is needed to understand numerical behavior. Second, neuromorphic approaches provide a new class of learning algorithms that must scale while ensuring accurate prediction. Coordinated hardware-software research is needed to demonstrate the potential of spiking neural networks (software) and to accelerate computation for artificial neural networks (hardware). Third, multi-agent approaches are well suited for learning the dynamics within complex systems and social settings. These approaches analyze agent objectives, infer agents' strategies, and characterize system dynamics as agents implement their strategies. Multi-agent systems can leverage algorithmic economics and game theory, but research is needed to operationalize the theory for practical systems.

Contextual analysis provides information about an environment by analyzing sensed data. These applications are characterized by real-time computation on streaming data. For example, a mobile device (e.g., phone or tablet) senses an audio stream to perform language transcription and translation, or it senses a video stream to perform face detection and recognition. The analysis fits a data flow model, motivating research to organize computation into stages for pipeline parallelism and across machines for data parallelism. Computation must be organized to exploit the inherent parallelism in an application's data flow graph, to accommodate constraints from mobile hardware limitations and server communication costs, and to meet real-time objectives. The growing divide between datacenter and mobile hardware raises new questions. How do we design platforms to support contextual analysis and intelligent personal assistants? How do we distribute (pieces of) computation across diverse platforms? How do we balance computation and communication costs?

Finally, scientific computing requires simulation with increasing fidelity. XPS research is needed to provide increasingly capable hardware platforms for computational physics and biology. In physics, parallel and scalable systems reduce the number of required modeling assumptions in simulations for turbulent reacting flows, computational material science, and other multi-scale problems. As computational capability increases, simulations will play integrated roles in engineering design and uncertainty quantification will become tractable for complex multiphysics, multi-scale systems. In

biology, scalability increases throughput for medical imaging---image processing, feature extraction, machine learning---and genetic analysis---genomic sequencing and regularity network analysis. Increased throughput for these applications will translate into advances in neuroscience, drug discovery, and personalized medicine.

2.3 Algorithms

XPS research is needed to produce new algorithms that produce the same application solution with less work, which is measured in terms of the number of operations performed or the amount of data moved. Along with technologies and architectures, algorithms are a key determinant in application performance and efficiency. Algorithms research must be coordinated with advances in architecture and systems. Because efficient algorithms avoid communication and data movement, their development must be cognizant of cache and memory technologies, of locality and the costs of non-uniform memory access, and of consistency models and coherence protocols.

Whereas algorithms have long been studied for key kernels in scientific computing, XPS research is needed to develop scalable algorithms for big data analytics. In some cases, scientific computing and statistical machine learning share kernels such that algorithmic advances for one would benefit the other. For example, collaborative filtering factors a sparse user-item association matrix to create a recommender system (e.g., book or movie suggestions). More generally, fast and distributed libraries for sparse linear algebra are likely to benefit both scientific computing and statistical machine learning. Although libraries (e.g., R, Mahout, MLlib) have democratized access to machine learning, XPS research is needed to increase parallelism and scalability on diverse hardware platforms (e.g., GPUs, datacenters, etc.)

Algorithms in sparse linear algebra complement those for irregular graphs. Indeed, graph analytics is a vibrant research area that has produced easily deployed software frameworks and run-time systems. Statistical machine learning on graphs requires new algorithms that accommodate sparsity and irregularity. XPS research is needed to balance computation and communication, to balance consistency and parallelism, and to apply abstractions for graph computation in multiple domain (e.g., biological networks, social networks, diffusion and contagion, etc.).

Finally, research is needed in algorithms for approximate and probabilistic computing. Approximate computing relaxes the requirements for precise results and probabilistic computing relies on sampled data. Approximate algorithms accommodate imprecision in emerging technologies, such as scaled CMOS, non-volatile memory, and nanophotonics. Probabilistic algorithms accommodate data sources from sensors

integrated with the physical world. XPS research is needed to develop approximate and probabilistic algorithms that produce a sufficiently precise answer while using much less time and energy.

2.4 Programming Systems

The role of programming language (PL) techniques is a core component of XPS, since new systems need solid programming models and tools. However, one could argue that the PL community has traditionally been inwards-focused. However, this is starting to change with significant advances in applying PL technology (verification, model checking, DSLs) to more general problems. We bring this up because future versions of the XPS program should directly encourage the participation of the PL community.

Programming languages and systems are core components of an XPS research portfolio as new algorithms, architectures and systems demand corresponding advances in programming models and tools. Hardware diversity is increasing as systems deploy a mix of heterogeneous general-purpose cores, graphics processors, specialized accelerators, and reconfigurable logic. Such diversity will exist across all computational scales --- from embedded devices to warehouse-scale datacenters. Diverse hardware platforms complicate the development of large software systems, which are increasingly written in a heterogeneous mix of languages and run-time systems. XPS research is needed in programming systems to enhance programmer productivity, improve performance portability, and harness these diverse computational resources. Research is also needed in robust compiler analysis and optimization for languages that have not traditionally received such support, including dynamic scripting and functional languages. These languages will contribute to an already rich ecosystem of frameworks for parallel programming and distributed computing. Moreover, research is needed in software optimization that crosses program boundaries to consider distributed tasks that comprise a datacenter job.

Domain specific languages (DSLs) and approaches in generative programming will require further research to deliver important tools for real world software development. The biggest DSL success, SQL, defines a restricted language that permits elaborate query optimization. Research is needed in generative programming in which high-level languages provide generality and abstraction, allowing programmers to build large systems from simple and versatile parts. Parallel and scalable systems studied by XPS researchers will increasingly combine heterogeneous parts to produce a more capable whole. Moreover, given increasing generality, compiler research is needed to translate high-level programs into efficient code. Balancing the power of expressive languages against the efficiency of code generation continues to be a fundamental challenge.

2.5 Evaluation Strategies

Research in applications, algorithms, and programming systems suffer from a lack of clearly defined figures of merit. Moreover, researchers often confront tradeoffs between subjective / qualitative measures (e.g., programmer productivity and code portability) and objective / quantitative measures (e.g., performance and energy efficiency). XPS research is needed in evaluation methodologies as illustrated in two representative research areas --- hardware specialization and programming systems.

First, computation on specialized or accelerated hardware can be 10-100x more energy-efficient than the same computation on general-purpose hardware. These quantitative benefits are offset by qualitative costs in programmability and design. How should researchers balance objective measures of efficiency and subjective measures of human effort? Second, programming systems can improve productivity but how should productivity be measured? For example, computation on graphics processors can be 10x faster than the same computation on general-purpose processors. How should researchers balance the objective measures of performance and subjective measures of human effort?

3 Architectures and Systems

The hardware substrates for parallel and scalable computing are experiencing a revolution, driven by the impending end of Moore's Law and the proliferation of heterogeneity. Without Dennard scaling to provide smaller, faster, and more efficient transistors, responsibility for continued advances in computing capability falls on architecture and system design. Research is needed in methodologies that generate and optimize new hardware architectures while reducing design effort and suppressing non-recurring engineering costs. As design becomes less difficult, architectures and systems will become increasingly heterogeneous, posing significant challenges in systems management, resource allocation, and task scheduling. XPS research is needed to enable design for manageability---the idea of anticipating management risk at design time.

3.1 Design and Optimization

Falling costs in hardware design and fabrication would constitute a major disruption in computing. Reduced costs would blur the conventional wisdom that divides hardware and software --- computation in software is less expensive than in hardware. Reduced costs are driven by two trends. First, fabrication costs will fall as the economic case

behind Moore's Law falters and foundries increasingly fabricate designs on older technology nodes (e.g., 22nm rather than 7nm process). At this inflection point, a foundry's capital costs for older nodes will be amortized over a period that is much longer than the 18-24 months dictated by Moore's Law. Architects will bear smaller shares of the foundry's capital costs and see lower fabrication costs. Second, design costs will fall as high-level synthesis tools become more effective when compiling software into hardware designs. XPS research is needed to accelerate advances in high-level synthesis and their effective application in hardware design for diverse application domains.

As falling costs make many designs feasible, design-space exploration becomes increasingly important. Hardware parameters define an enormous design space. Coordinated hardware-software design further increases the challenge. XPS research is needed in methodologies that optimize designs and identify Pareto optima in terms of varied metrics---performance, energy efficiency, chip area, programmability, etc. Emerging computational paradigms, such as approximate and probabilistic computing, will further extend the list of optimization criteria to include solution quality. In hardware design, research is needed in architectural modeling, component optimization, and system composition. In software tuning, research is needed in algorithm selection, data structure tuning, and parallelism management. Finally, XPS research should emphasize hardware-software co-design.

Beyond the design of individual hardware components, XPS research is needed to organize diverse components (e.g., processors, memories, interconnects with varied technologies) into coherent systems. Emerging technologies and integration strategies make system balance a first-order design objective. For example, how do processors communicate with memories? What data bandwidth is required from memories and their interfaces given the instruction throughput from a processor design? Such questions become increasingly difficult in an era when coordinating advances in accelerated and heterogeneous computing with emerging memories and interconnects.

3.2 Management and Risk

Systems management is required to deliver application objectives. However, as less expensive design produces many diverse hardware components, systems management becomes far more challenging. For example, ensuring service quality in a datacenter with homogeneous processors is already difficult, and doing so in a datacenter with heterogeneous processors and accelerators is even more so. XPS research is needed to manage hardware deployed in parallel and scalable systems. Specifically, resource allocation and task scheduling must be aware of underlying hardware capabilities,

allowing them to account for heterogeneity and risks to performance from contention, faults, and variability. In the pursuit of these goals, allocation and scheduling will benefit from machine learning (e.g., for modeling and prediction) and economic mechanisms (e.g., for contention management). Allocation and scheduling must define the objective, whether it be performance, fairness, or a combination of the two.

XPS research is needed in management mechanisms to mitigate contention for shared resources. Today's scalable and distributed systems consist of large servers with many cores, large caches, and abundant memory. Parallel tasks will compete for these resources, and systems must arbitrate access to achieve objectives in service quality, latency, throughput, and fairness. Moreover, new types of contention arise in heterogeneous systems. What happens when tasks prefer the same type of hardware? Should we allocate to maximize task throughput? Or should we allocate to minimize the penalties from receiving sub-optimal alternatives? These question pose even greater challenges when coordinating the allocation of multiple resources (e.g., processors and memory).

Hardware faults produce a second source of management risk. As systems scale, the probability of failure (hard or soft), degradation, and aging increases. XPS research is needed in resilient architectures and systems that can recover autonomously from failure. In the future, advances in reconfigurable computing could provide mechanisms for fault recovery. Moreover, advances in neuromorphic architectures could produce resilient platforms that implicitly mask faults due to the nature of approximate computing. Finally, the emergence of fast, persistent memories could qualitatively improve data durability and availability by reducing the window of vulnerability from node failures and power loss.

At scale, performance outliers and tail behaviors produce a third source of management risk. Outliers that occur with low probabilities will arise in large numbers within a datacenter that deploys tens of thousands of processors for millions of tasks. XPS research is needed to characterize the tail distribution and develop management strategies that mitigate their effect on service quality. Research is needed in profiling mechanisms that quantify the frequency of outliers, in modeling methodologies that reveal root causes and predict their occurrences as well as in system architectures that accelerate lagging tasks to meet performance objectives.

3.3 Heterogeneity and Parallelism

XPS research in design and management must address pervasive heterogeneity. In hardware, systems architects are turning to heterogeneous mixes of general-purpose

processor cores to balance performance and efficiency. For example, datacenters might deploy a mix of low-power processors for common, simple queries and high-performance processors for the rare, complex ones. Similarly, mobile platforms might deploy low-power cores for normal operation but active high-performance ones to accelerate computation occasionally. Beyond mixes of general-purpose cores, heterogeneous systems increasingly deploy programmable data-parallel accelerators (e.g., general-purpose graphics processing units or GPGPUs), reconfigurable logic (e.g., field-programmable gate arrays or FPGAs), neuromorphic accelerators, and other specialized capability. XPS research is needed to optimize the composition of heterogeneous systems and to steer computation to the best suited component.

Heterogeneous systems give rise to a host of new software frameworks and run-time systems. XPS research is needed to integrate heterogeneous programming frameworks. Within a node, programmers increasingly deploy a mix of general-purpose parallel frameworks (e.g., POSIX threads, Cilk++, Intel TBB, OpenMP) and data-parallel frameworks (e.g., CUDA, OpenCL, OpenMP). Within a distributed system, programmers increasingly navigate diverse frameworks that are tailored for specific application domains. Since MapReduce separated the programming model and run-time system for versatile task management, similar frameworks have been proposed for more sophisticated computation (e.g., machine learning, graph analytics). Future large, distributed systems will deploy a mix of frameworks that will require hierarchical and coordinated management. For example, how should we divide resources between MapReduce and Spark? How should we divide resources between tasks within each framework?

Beyond hardware and software heterogeneity, systems increasingly encounter heterogeneity from hardware-software interaction. Data movement and its non-uniform costs are a prominent example of such heterogeneity. XPS research is needed to manage locality and navigate the varied paths to data. Within a distributed system, task schedulers should be cognizant of locality and schedule tasks to manage data movement to and from the file system. Similar questions are increasingly relevant within a datacenter node. In the future, servers will include high-capacity memory systems and multiple paths to data that mix parallel buses and serial links for processor-processor, processor-memory, and buffer-memory communication. XPS research is needed to place data and schedule tasks that ensure performance despite heterogeneous and non-uniform memory access.

4 Emerging Technologies

Two related forces have driven technology scaling. First, in 1965, Gordon Moore observed that component integration reduced cost. This observation made the economic case for smaller transistors and became the basis for Moore's Law, which described the doubling of transistor density every 18-24 months. Second, in 1974, Robert Dennard provided the scaling guidelines that would implement Moore's Law. Dennard scaling reduces not only transistor dimensions but also their operating parameters (i.e., voltage and current) to ensure constant power density---transistors would dissipate less power even as they occupied less area.

Dennard scaling ended in the early 2000s, yet the economic case behind Moore's Law has persisted. The semiconductor industry has relied on a series of unique strategies to advance process technologies to the next node (e.g., strained silicon to increase carrier mobility, multi-gate FinFETs to increase control over the gate). However, these advanced technology nodes are increasingly expensive, especially when accounting for process variations and their effect on yield. Computer systems and architectures are rapidly approaching an inflection point, and research is needed to anticipate the end of Moore's Law.

XPS research should focus on architectures that deliver the performance and power efficiency that was once delivered by technology scaling. Moreover, it should explore emerging technologies and assess their implications for architecture, systems, and applications. Research in emerging technologies must be tightly integrated with research in systems and applications---a technology will be forever emerging without a compelling application to produce economies of scale.

4.1 Beyond Moore's Law

Historical gains in performance and energy efficiency have been driven by technology scaling and architectural design. In an era of constrained scaling, continued gains require research in architecture, systems, and applications. Domain-specific design can produce accelerators with customized datapaths, memories, and control that improve performance and efficiency by orders of magnitude (e.g., 10-100x) over general-purpose solutions. A general-purpose processor consumes 10% of its energy for an instruction and 90% of it on overhead, such as data movement. A specialized processor tailors memory and data movement to complete much more work with much less overhead. When the costs of supplying instructions and data are amortized over hundreds of computational operations, performance and efficiency improve by orders of magnitude.

XPS research is needed in methodologies for accelerator design. Although the potential gains for performance and efficiency have been demonstrated, design methods are needed to systematically produce these gains for diverse applications. Design methodologies are needed in high-level synthesis to generate hardware designs from software specifications, in performance and power modeling to assess the merits of each design, and in design-space exploration to optimize figures of merit subject to budgetary constraints. Accelerators, which use specialization not process technology to deliver efficiency, may be fabricated on older, less expensive technology nodes (e.g., 22nm). As each technology node finds greater application, a foundry's capital costs will be amortized over longer schedules and architects will benefit from lower fabrication costs.

4.2 Alternative Technologies

Silicon and complementary metal-oxide semiconductors (CMOS) may remain the most cost-effective technology for years to come. Yet, alternative technologies are emerging to offer qualitatively new capabilities. For a compelling set of applications, these technologies could revolutionize computing by eliminating today's pressing technology constraints. XPS research is needed to identify and enable applications that exploit new technologies in memory, communication, and packaging.

Emerging non-volatile memories promise to transform computing with (relatively) competitive performance, density, and scalability. Yet identifying the specific technology that will succeed is challenging---phase-change memory, magnetoresistive RAM, and memristors continue to show promise. Instead of picking winners in this race for commercially viable technologies, XPS research should develop parameterize models for varied technology scenarios, identify the fundamental properties of these "storage class memories," and examine the implications for architectures, systems, and applications. How should application programmers and storage systems exploit fast persistence? How can resistive memories support new paradigms in computing (e.g., approximate, neuromorphic)?

Novel communication networks and component packaging strategies are needed to support sophisticated and irregular interfaces between increasingly diverse components. Today's system architect confronts a mix of communication technologies---parallel buses, serial links, through-silicon vias, optical waveguides. These technologies reflect a mix of integration strategies for today's systems---buffered communication between chipsets, 2.5D packages with silicon interposers, and 3D packages with through-silicon vias. XPS research is needed to organize these

technologies into balanced interconnection networks that reflect processor and memory demands for data movement. Furthermore, research is needed in versatile communication and command protocols that support varied technologies and interfaces.

Emerging technologies foreshadow an era of heterogeneous capability with interfaces that have not yet been defined. Whereas standardized technologies in memories and network have encouraged ever greater integration of memory and network controllers within the processor, emerging technologies may encourage a return to dis-integrated chip sets that allow system architects to mix and match capabilities, according to application demands. XPS research is needed in designing components with emerging technologies, defining component interfaces, and organizing components into modular and coherent systems.

4.3 Cross-Cutting Challenges

XPS research should architect systems that exploit emerging technologies and evaluate them holistically with a particular emphasis on three figures of merit --- balance, power, and reliability. First, system balance ensures that emerging technologies improve capability without introducing bottlenecks. The notion of balance was first introduced in the 1980s to reason about the relationships between instruction throughput in the processor, data bandwidth provided by storage, and the amount of data required for each instruction. Imbalances between data demanded by the processor and data supplied by the storage system would be corrected by local memory caches. Today's system architects face the same fundamental challenges with a plethora of new technologies and capabilities.

Second, power efficiency and its related figures of merit --- energy efficiency and thermal management --- require new technologies that fundamentally change existing performance and power tradeoffs. For example, emerging memories that are non-volatile and bandwidth constrained can improve energy proportionality in the memory system by eliminating refresh in the memory core and synchronization circuitry at the memory interfaces. Technologies that increase throughput and bandwidth pose particular challenges --- power increases with capability posing thermal difficulties. XPS research is needed in methodologies that balance performance and power to produce Pareto optimal designs given emerging technologies for memory cells, memory interfaces, packaging and integration, and interconnection networks.

Third, reliability is a severe challenge when existing technologies are scaled or emerging technologies are adopted. System architects should design for resilience in the face of process variations during fabrication and endurance limitations at run time.

XPS research is needed to compute in the presence of faults with hardware and software support for reconfiguration and error correction. Emerging technologies (e.g., resistive memories) and accompanying computational paradigms (e.g., neuromorphic computing) may be naturally resilient to faults. The stochastic nature of resistive memories supports analog weights in a neural network. XPS research is needed to manage approximation across multiple abstractions layers, including technologies, architectures, systems, and software.

5 Research Methodologies and Risks

The strength of the XPS research program has been its interdisciplinary research that spans the hardware-software stack. Effective cross-stack research will require the continued emphasis on collaborative teams that integrate diverse perspectives and on research methodologies that encourage innovation from researchers in adjacent stack layers.

5.1 Interdisciplinary Research

Vertically integrated teams are essential for interdisciplinary work. Teams that tightly link researchers across two or more layers (e,.g., applications, systems, architectures, and technologies) in the hardware-software stack are more likely to make deep and lasting contributions to computing. Such teams are particularly important as abstraction layers evolve in two directions --- bottom-up (driven by foundations and principles) and top-down (driven by applications). Vertically integrated teams are more likely to discover consistent and coherent outcomes from these simultaneous evolutions. As abstractions evolve, researchers must make design decisions explicit --- What has been hidden? What has been exposed? What abstractions are porous? XPS research should break and re-define abstractions, but researchers should know exactly which abstractions are broken and argue persuasively for the redefined abstractions.

Furthermore, interdisciplinary research requires methodologies and tools that facilitate innovation across abstraction layers. XPS research should distribute research artifacts openly and encourage researchers in other communities to leverage these artifacts to make complementary advances. Artifacts should be distributed across abstraction layers to improve community adoption and increase research impact. Further research is needed in workload characterization and benchmarking to identify representative applications for a new era of computing (e.g,. internet of things and big data). The behavior of these applications may depend on data inputs, and realistic datasets are needed to drive research. Finally, new technologies and architectures demand

parameterized simulators and register transfer language (RTL) implementations that broaden the community and permit further research across hardware-software interfaces.

5.2 Risks and Threats

XPS research encounters several threats to success, and researchers should mitigate these risks. First, researchers must remember Amdahl's Law and its equivalent for figures of merit beyond performance. Focusing on parallelism alone for performance and energy efficiency will produce diminishing marginal returns, especially since parallelism has become pervasive in the past decade --- witness the spread of data parallelism in general-purpose computing on graphics processors or the spread of task parallelism in programming models and run-time systems for distributed computing. XPS research is needed to further complementary advances in reconfigurable, approximate, or domain-specific computing. Similarly, XPS research is needed to look beyond processors to consider memory, storage, and interconnect. As processors become increasingly capable and efficient, the components that hold and move data will introduce performance and energy-efficiency bottlenecks.

Second, cross-stack and interdisciplinary research requires assumptions about next-generation applications and emerging technologies. However, research should be parameterized to reflect uncertainty and rapid transitions in computing. Researchers risk placing losing bets on applications and technologies as industrial development and economic incentives shift roadmaps. While XPS research should be motivated by projections into the future, it should also be structured to produce basic technologies that enable previously inconceivable applications --- even if projections prove incorrect, projects should produce fundamental insight.

Finally, the role of industry poses a risk to the relevance of academic research. Academic research continues to be invaluable for several reasons --- (i) performing fundamental research without the constraints imposed by product roadmaps; (ii) studying questions that span multiple domains, disciplines, and companies; and (iii) training next-generation researchers and scientists in computing. However, academic research would benefit from broader interfaces with industry, which may have significant datasets and large systems that would increase the impact of academic projects. XPS researchers should collaborate with industry to better understand their constraints, to evaluate applications and systems at scale, and to better train the next generation of computer scientists.

6 Closing Thoughts

The future of XPS research is bright. The National Science Foundation's XPS research program has fostered an interdisciplinary community of researchers who seek to answer big and difficult questions in scalable computing. As demonstrated in this report, the XPS community prioritizes cross-cutting research for scalable computing. Indeed, the XPS acronym could be repurposed to concisely describe this community's priorities --- Cross-cutting (X) Pathways (P) to Scalability (S).

Appendix A Workshop Overview

A.1 Workshop Overview

The Exploiting Parallelism and Scalability (XPS) program aims to support groundbreaking research leading to a new era of parallel computing. Achieving the needed breakthroughs will require a collaborative effort among researchers representing all areas --- from servers and applications down to the microarchitecture and emerging technologies --- and will be built on new concepts, theories, and foundational principles. New approaches to achieve scalable performance and usability need new abstract models and algorithms, new programming models and languages, new hardware architectures, compilers, operating systems, and run-time systems, and must exploit domain-specific knowledge. Research is also needed on energy efficiency, communication efficiency, and on enabling the division of effort between edge devices and clouds.

A.2 PI Meeting

XPS investigators are invited to attend this workshop, which will encompass discussions about the past, present, and future of exploiting parallelism and scalability (XPS). Specifically, the workshop will review current projects and will look forward in shaping the future of XPS. To this end, the workshop seeks to engage in “big picture” discussions in XPS and identify priorities for future research and solicitations.

A.3 Workshop Dates and Location

1-2 June 2015
Virginia Tech Research Center -- Arlington
900 N. Glebe Road, Arlington, VA 22203

A.4 Steering Committee and Organizers

Wu Feng, Virginia Tech (Co-Chair)
Benjamin Lee, Duke University (Co-Chair)
Kunal Agrawal, Washington University in St. Louis
Rastislav Bodik, University of Washington
Luis Ceze, University of Washington
Lingjia Tang, University of Michigan

Appendix B Workshop Themes

B.1 Cross-Cutting Approaches

In order to fully exploit the power of current and emerging technologies, research is needed to revisit assumptions underlying traditional approaches --- to applications, data management, data mining and machine learning systems, programming languages, compilers, run-time systems, virtual machines, operating systems, architectures, and hardware/microarchitectures -- in light of current and future heterogeneous parallel systems. A successful approach should be a collaboration that explores new holistic approaches to parallelism and cross-layer design.

- New abstractions, models, and software systems that expose fundamental attributes, such as energy use, and communication costs, across all layers and that are portable across different platforms and architectural generations.
- New software and system architectures that are designed for exploitable locality, with parallelism and communication efficiency to minimize energy use, and using on-chip and chip-to-chip communication achieving low latency, high bandwidth, and power efficiency.
- New methods and metrics for evaluating, verifying and validating correctness, reliability, resilience, performance, and scalability of concurrent, parallel, and heterogeneous systems.
- Run-time systems to manage parallelism, memory allocation, synchronization, communication, I/O, data placement, and energy usage.
- Extracting general principles that can drive the future generation of computing architectures and tools with a focus on scalability, reliability, robustness, and verifiability.
- Exploration of tradeoffs addressing an optimized separation of concerns across layers. Which problems should be handled by which layers? What information, using which abstractions, must flow between the layers to achieve optimal performance? Which aspects of system design can be automated and what is the optimal use of costly human ingenuity?
- Cross-layer issues related to the support of large-scale distributed computational science applications.

B.2 Domain-Specific Design

Research is needed on foundational techniques for exploiting domain and application-specific knowledge to improve programmability, reliability, and scalable performance. Topics include, but are not limited to:

- Parallel domain-specific languages, including query languages, that provide both high-level programming models for domain experts and high performance across a range of parallel platforms, such as GPUs, SMPs, and clusters.
- Program synthesis tools that generate efficient parallel codes and/or query processing plans from high-level problem descriptions using domain-specific knowledge. Approaches might include optimizations based on mathematical and/or statistical reasoning, set theory, logic, auto-vectorization techniques that exploit domain-specific properties, and auto-tuning techniques.
- Hardware-software co-design for domain-specific applications that pushes performance and energy efficiency while reducing cost, overhead, and inefficiencies.
- Integrated data management paradigms harnessing parallelism and concurrency; the entire data path from data generation to transmission, storage, access, use, maintenance, analytics, and to eventual archiving or destruction, is in scope.
- Work that generalizes the approach of exploiting domain-specific knowledge, such as tools, frameworks, and libraries that support the development of domain-specific solutions to computational and data management problems and are integrated with domain science.
- Novel approaches suitable for scientific application frameworks addressing domain-specific mapping of parallelism onto a variety of parallel computational models and scales.

B.3 Foundational Principles

Research on foundational principles should engender a paradigm shift in the ways one conceives, develops, analyzes, and uses parallel algorithms, languages, and concurrency. Foundational research should be guided by crucial design principles and constraints impacting these principles. Topics include, but are not limited to:

- New computational models that free algorithm designers and programmers from many low-level details of specific parallel hardware while supporting the expression of properties of a desired computation that allows maximum parallel performance. Models should be simple enough to understand and use, have solid semantic foundations, and guide algorithm design choices for diverse parallel platforms.
- Algorithms and algorithmic paradigms that simultaneously allow reasoning about correctness and parallel performance, lead to provable performance guarantees, and allow optimizing for various resources, including energy and data movement

(both memory hierarchy and communication bandwidth) as well as parallel work and running time.

- New programming languages, program logics, type theories, and language mechanisms that support new computational and data models, raise the level of abstraction, and lower the barrier of entry for parallel and concurrent programming. Parallel and concurrent languages that have programmability, verifiability, and scalable performance as design goals. Of particular interest are languages that abstract away from the traditional imperative programming model found in most sequential programming languages.
- Compilers and techniques, including certification, for mapping high-level parallel languages and language mechanisms to efficient low-level, platform-specific code.
- Development of interfaces to express parallelism at a higher level while being able to express and analyze locality, communication, and other parameters that affect performance and scalability.
- New data models, query languages, and query optimization techniques that support large data sets and parallel processing for database, data mining, and machine learning queries.
- Novel approaches to designing and analyzing heterogeneous hardware, programmable logic, and accelerators, and to hardware support for programmability (e.g., transactional memory) and reliability (e.g.,recovery blocks).

B.4 Scalable Distributed Architectures

Large-scale heterogeneous distributed systems (e.g., the web, grid, cloud) have become commonplace in both general purpose and scientific contexts. With the increased prominence of smart phones, tablets, and other types of edge devices, users expect these systems to be robust, reliable, safe and efficient. At the same time, new applications leveraging these platforms require a rich environment that enables sensing and computing with diverse distributed data, along with communication among and between these systems and the elements that comprise them. Research supporting the science and design of these extensible distributed systems, particularly the components and programming of highly parallel and scalable distributed architectures, will enable the many "smart" technologies and infrastructures of the future. Topics include, but are not limited to:

- Novel approaches enabling heterogeneous edge devices - with constraints such as low energy use, tight form factors, tight time constraints, adequate

computational and data management capacity, and low cost - to collaborate in delivering computation-intensive applications utilizing distributed data.

- Runtime platforms and virtualization tools that allow programs to divide effort among portable platforms and large-scale compute and data resources while responding dynamically to changes in reliability and energy efficiency. Possible questions include: How should computation be mapped onto the elements of large-scale distributed systems? How can system architecture help preserve privacy by giving users more control over their data?
- Research that enables conventionally-trained engineers to program computing systems extending across wide geographic scales, taking advantage of highly parallel and distributed environments while simultaneously exhibiting resilience to significant amounts of component and communication failures. Such research may be based on novel hardware support, programming abstractions, algorithms, storage systems, middleware, operating systems, or data management, data mining systems, and machine learning systems.

Appendix C Workshop Participants

C.1 Cross-Cutting Approaches

- **Luis Ceze, University of Washington, Session Leader**
- Abhishek Bhattacharjee, Rutgers University
- Alvin Lebeck, Duke University
- Andreas Gerstlauer, University of Texas, Austin
- Byunghyun Jang, The University of Mississippi
- David Wentzlaff, Princeton University
- Hai Li, University of Pittsburgh
- Hyesoon Kim, Georgia Institute of Technology
- Mark Oskin, University of Washington
- Michael Ferdman, Stony Brook University
- Minsui Choi, Missouri University of Science and Technology
- Sherief Reda, Brown University
- Rob Johnson, Stony Brook University

C.2 Domain-Specific Design

- **Rastislav Bodik, University of California, Berkeley, Session Leader**
- Christopher Batten, Cornell University
- David Brooks, Harvard University
- Diego Donzis, Texas A&M University
- Geoffrey Fox, Indiana University
- George Biros, University of Texas, Austin
- Kamesh Madduri, The Pennsylvania State University
- Kunle Olukotun, Stanford University
- Lawrence Rauchwerger, Texas A&M University
- Mahmut Kandemir, The Pennsylvania State University
- Matt Might and James Sutherland, University of Utah
- Nikos Chrisochoides, Old Dominion University
- Qinru Qiu, Syracuse University
- Zhiru Zhang, Cornell University

C.3 Foundational Principles

- **Kunal Agrawal, Washington University in St. Louis, Session Leader**
- Andrew Lenharth, University of Texas, Austin
- Ganesh Gopalakrishnan, University of Utah

- Guang Gao, University of Delaware
- Gul Agha, University of Illinois
- Jonathan Appavoo, Boston University
- Jose Renau, University of California, Santa Cruz
- Joseph Devietti, University of Pennsylvania
- Jun Wang, University of Central Florida
- Milind Kulkarni, Purdue University
- Nathaniel Cady, SUNY Polytechnic Institute
- Ryan Newton, Indiana University
- Scott Mahlke, University of Michigan
- Sumit Jha, University of Central Florida
- Ulya Karpuzcu, University of Minnesota

C.4 Scalable Distributed Architectures

- **Lingjia Tang, University of Michigan, Session Leader**
- Andy Pavlo, Carnegie Mellon University
- Haryadi Gunawi, University of Chicago
- Jason Mars, University of Michigan
- Michael Swift, University of Wisconsin
- Omer Khan, University of Connecticut
- Peter Varman, Rice University
- Richard Han, University of Colorado, Boulder
- Sandhya Dwarkadas, University of Rochester
- Shan Lu, University of Chicago
- Srini Devadas, Massachusetts Institute of Technology
- Stephen Freund, Williams College
- Tao Wei, University of Rhode Island
- Yuqing Wu, Pomona College

Appendix D Workshop Program

1 June 2015

8:00 -- 8:30AM	Breakfast
8:30 -- 9:00AM	Overview and Agenda
9:00 -- 10:15AM	Project Reports (Cross-Cutting Approaches) Guang Gao, Haryadi Gunawi, Hyesoon Kim, Peter Varman, Sherief Reda, Minsui Choi Project Reports (Foundational Principles) Jonathan Appavoo, Joseph Devietti, Nathaniel Cady, Ryan Newton, Ulya Karpuzcu, Jose Renau
10:15 -- 10:45AM	Break
10:45 -- 12:00PM	Panel (Applications) Milind Kulkarni, Rastislav Bodik, Gul Agha, Kunle Olukotun, Lawrence Rauchwerger
12:00 -- 1:00PM	Lunch
1:00 -- 2:00PM	Keynote: Algorithms and architectures for scalable N-body methods George Biros
2:00 -- 3:15PM	Discussion Sessions
3:15 -- 3:30PM	Break
3:30 -- 5:00PM	Discussion Sessions
5:00 -- 5:30PM	First Day's Report
5:30 -- 7:00PM	Reception and Poster Session

2 June 2015

8:00 -- 8:30AM	Breakfast
8:30 -- 9:00AM	Overview and Agenda
9:00 -- 10:15AM	Project Reports (Domain-Specific Design) Christopher Batten, Diego Donzis, Matt Might Nikos Chrisochoides, Qinru Qiu, Geoffrey Fox, Kamesh Madduri Project Reports (Scalable Distributed Architectures) David Wentzlaff, Richard Han, Shan Lu, Stephen Freund, Andy Pavlo, Yuqing Wu
10:15 -- 10:45AM	Break
10:45 -- 12:00PM	Panel (Systems) Srini Devadas, Alvin Lebeck, Mark Oskin Sandhya Dwarkadas, Michael Swift
12:00 -- 1:00PM	Lunch
1:00 -- 2:00PM	Keynote: Addressing the Computing Technology-Capability Gap: The Coming Golden Age of Design via Specialization & Parallelism. David Brooks
2:00 -- 3:15PM	Discussion Sessions
3:15 -- 3:30PM	Break
3:30 -- 4:15PM	Prepare for Out-Brief
4:15 -- 5:00PM	Workshop Report