Systems and Applications Challenges for the Emerging Bazaar of Accelerators Workshop Report

June 2024

The workshop was supported by the Computing Community Consortium through the National Science Foundation under Grants No. 1734706 and 2300842.
Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Systems and Applications Challenges for the Emerging Bazaar of Accelerators Workshop Report

Workshop
August 9-10, 2023

Workshop Organized by:

Catherine Schuman, University of Tennessee, Knoxville
John Shalf, Lawrence Berkeley National Laboratory
Thomas Conte, Georgia Institute of Technology

With Support From:

Catherine Gill, Computing Community Consortium (CCC)
Haley Griffin, Computing Community Consortium (CCC)
Maddy Hunter, Computing Community Consortium (CCC)
Ann Schwartz, Computing Community Consortium (CCC)

How to Site this Report:

Schuman, C., Shalf, J., & Conte, T. (2024). *Systems and Applications Challenges for the Emerging Bazaar of Accelerators* (Report). Computing Community Consortium.

CCC
Computing Community Consortium
Catalyst

Executive Summary

With the traditional definition of Moore's law waning, future performance gains are increasingly dependent upon architectural specialization, leading to ubiquitous and heterogeneous accelerators. This shift fundamentally changes the paradigm for programming and computation across all levels of the compute stack, and necessitates major changes in our fundamental research programs to prepare for the future of computing. In particular, these fundamental changes need us as the computing community to rethink the way we research and teach computing. Funding agencies also need to rethink the mechanisms for funding research in this space, particularly to encourage more cross-stack research programs. The way that we measure performance in computing must also be modernized, as metrics such as speed are no longer the only metrics we should consider.

In our workshop, *Systems and Applications Challenges for the Emerging Bazaar of Accelerators*, we discussed the particular challenges associated with this future of computing in which there are ubiquitous and heterogeneous accelerators. Below are the summarized key findings and recommendations from the workshop, on which we will elaborate in following sections of the report.

Key Findings

- Benchmarking and performance metrics are ill-suited for evaluating heterogeneous compute environments.
- Data movement exceeds the cost of computation.
- Existing co-design methodologies are shallow and only co-optimize neighboring layers of the hard-ware/compute stack.
- New programming abstractions are needed. Cross-disciplinary funding and research efforts are needed.
- Generalists are required to enable successful co-design.

Recommendations

- To enable deep algorithm/hardware/software co-design, new funding mechanisms must be established to allow these types of research to be funded contemporaneously in a given project.
- Researchers must adopt a data-centric programming approach to limit data movement and increase performance.
- Standardized accelerator interfaces should be researched and adopted widely.
- In education, we need efforts across the pipeline (from K12 through continued workforce training).
- We need to focus on creating/educating generalists in addition to specialists.
- Lowering the barrier of entry in accelerator development and integration is essential for fostering innovation and broadening participation to advance computing technologies.
- Interdisciplinary collaboration needs to be thought of and funded differently than traditional research programs.
- Consensus on the best practices for evaluating accelerator systems must be reached.

Workshop Description

The workshop, held in August 2023, included 27 attendees from academia, government, and industry. The workshop had three focused sessions on the role of AI in the emerging bazaar of accelerators, software sustainability and programming environments for diverse accelerators, and diverse hardware integration challenges.

History and Context

The emerging challenge of the slowing of Moore's Law as we have known it has led to a dramatic slowdown in the rate of performance improvement for HPC systems. As an example of the real-world impact of this slowing of Moore's law, Figure 1 shows that there had been consistent 1000x improvement of HPC performance delivered every 11 years in the early days, but that rate has slowed down to less than 10x every 11 years. The burgeoning AI/ML market has responded to the Moore's Law slowdown with a plethora of heterogeneous accelerators and memories as a means of continuing performance growth through architecture specialization. The bottom line is that the approach that the computing community has depended upon to procure systems that deliver exponential performance improvements to the scientific and engineering users over the past 3 decades is failing.

Another key challenge is the increase in demand for computing, alongside the tremendous increase in energy consumption of computing. In particular, if we continue to depend on computing systems of today, global computing energy consumption is set to become a major component of the world's overall energy consumption, approaching the world's total energy production (Figure 2). Additionally, the demand for computing is rapidly outpacing supply (Figure 3). Based on these two trends, if we do not make a fundamental change in computing hardware, there will be tremendous environmental and economic consequences.

With the end of Dennard scaling in 2005 and the tapering of improvements derived exclusively from shrinking the size of transistors (aka "Moore's Law"), the performance of individual processing cores has ceased to experience significant enhancements with each successive generation. The industry has adopted a paradigm shift by embracing heterogenous acceleration to continue performance improvements, where different types of cores are integrated into a single processor chip, often with the inclusion of heterogeneous and diverse accelerators – mostly driven by the AI/ML market. The simultaneous escalation of core heterogeneity and memory diversity has created an intricate web of complexities. An urgent need arises for higher-level abstractions to shield application developers from this growing complexity and reduce the effort required to adapt codes to different computing platforms. The demand for performance portability directly correlates with the increasing heterogeneity of computing platforms. Despite notable software solutions, the growing complexity of parallelism and memory hierarchy demand higher-level abstractions for performance portable programming systems across diverse computing platforms. There is a coming crisis in computing where current practices for design of hardware, software, and application design will be up-ended by these trends that are driven by the need for continued performance growth and energy efficiency.

There are several fundamental challenges associated with integrating a bazaar of accelerators together, both at the

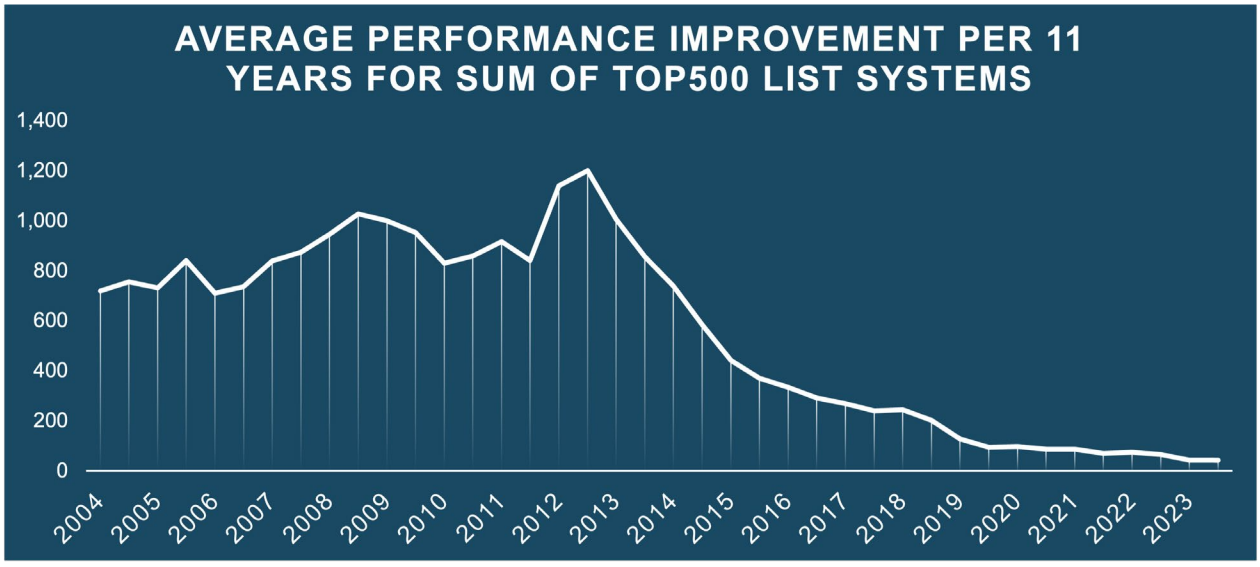

Figure 1: *The performance growth rate of HPC systems (as measured by LINPACK) has slowed from its historical growth rate of 1000x every 11 years down to just 3x for the same time period in 2023. (figure from Shalf Supercomputing 2024 Top500 BoF)*

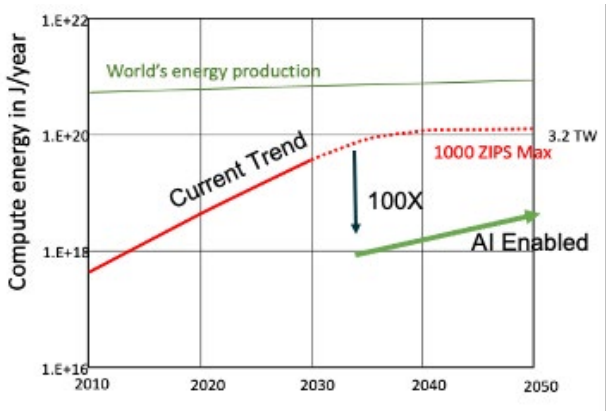

Figure 2: *Current trends in worldwide energy consumption of computing are pushing up against worldwide energy production limits with global consequences on the economy if this happens. (Ang, James et al. "Decadal Plan for Semiconductors." n.d. https://www.semiconductors.org/wp-content/uploads/2020/10/Decadal-Plan_Interim-Report.pdf.)*

hardware level and the software level. In hardware, a variety of emerging computing types (quantum, neuromorphic, analog, probabilistic), as well as non-CMOS devices to implement them are being used in the development of accelerators. Integration between non-CMOS and CMOS systems is non-trivial, as is packaging and manufacturing for these systems. Moreover, it is not clear what communication should look like with accelerators; some accelerators will be amenable to shared memory or shared storage, but others will be limited. As such, there is also a key challenge in determining how a set of diverse accelerators should ultimately be integrated together to form a complete hardware system, as well as how data should be moved throughout the heterogeneous system.

In software, different accelerators require fundamentally different ways of thinking about programming, which will require new programming languages and abstractions to be developed. This challenge has already arisen with diverse CPU and GPU-based systems; it will only become exponentially worse as more types of hardware accelerators are added.

A key challenge that also arises in the development of heterogeneous systems with bizarre accelerators is co-design. If hardware systems are developed in isolation without considering their potential future use as an accelerator, it will be near impossible to integrate those systems as-is into larger systems. Co-design across the compute stack, from the device and materials used to develop an accelerator all the way to the programming methodologies and programmers that will be using the accelerator in the larger system should be considered.

As new programming abstractions and methodologies are developed, and as the need for co-design across the whole computational stack continues, we will also need to rethink our educational systems to support these efforts. In particular, we will need to rethink how we teach and re-train programmers for these new heterogeneous compute environments. We will also need to consider the development of educational programs that support the training for co-design across the stack.

Finally, it is worth noting that the current funding structures often limit projects that truly span across the compute stack,

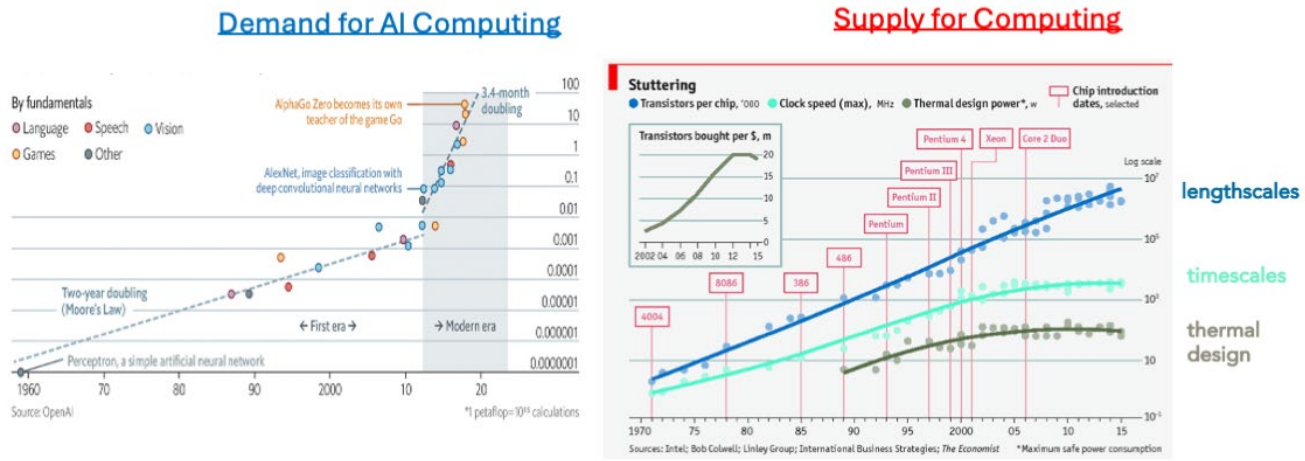

Figure 3: *Demand for computing compared to the supply for computing. Colwell, Bob. 2024. Economist.com. 2024. https://www.economist.com/sites/default/files/20200613_TQC666.png. and "Double, Double, Toil and Trouble." 2016. https://www.economist.com/sites/default/files/march_2016.pdf.)*

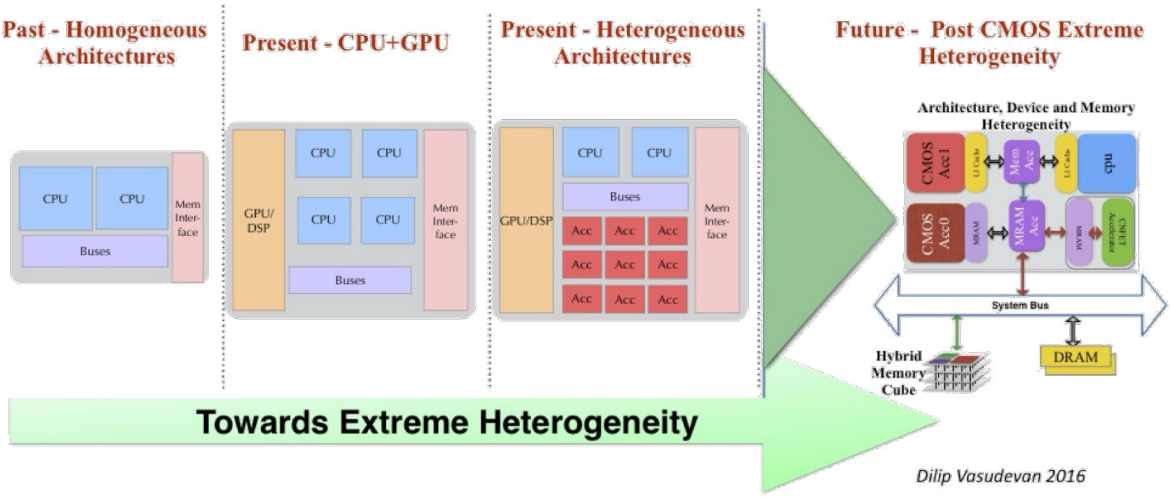

Figure 4: *The future of computing is increasingly dependent upon heterogeneous acceleration.* (William Chen and Dilip Vasudevan)

from devices and materials, to fabrication, to systems, to programming abstractions, to human computer interaction – all of which will be required for the future of computing with a bazaar of accelerators. Funding agencies should support broad, interdisciplinary programs that support teams that can do co-design efforts at this scale.

Overview of Major Findings

In response to the tapering of traditional performance improvements derived from Moore's Law, the broader computing industry at all scales has turned to heterogeneous acceleration to deliver continued performance growth through specialization. Traditional practices in hardware design, software, and general application design will become overwhelmed by the complexity of this emerging paradigm shift towards extreme heterogeneous acceleration. The findings of this workshop outline barriers and challenges that must be overcome in order to continue advancing the capabilities of modern accelerators in a post-Moore's law future.

- **Benchmarking and performance metrics are ill-suited for evaluating heterogeneous compute environments.** *If you can't measure something you can't improve it. Current performance metrics and computer benchmarking practice for measuring performance presumes a universal/general-purpose instruction processor, but we are rapidly evolving away from that reality.* ***The research community must fundamentally re-evaluate and re-invent benchmarking and performance metrics for a heterogeneous future.***

- **Data Movement exceeds the cost of computation.** *Successful specialization and heterogeneous acceleration will put even more pressure on data movement to be successful. The "memory wall" (coined in 1994) is just one manifestation of this broader observation about the increasing cost of data movement relative to computation.* ***Considerations about data movement are crucial to successful deployment of accelerators. Metrics for algorithm complexity will need to evolve from expressing computational complexity to consider also the data-movement complexity and cost.***

- **Existing co-design methodologies are shallow and only co-optimize neighboring layers of the hardware/compute stack.** *Hardware/software co-design has been an effective approach to managing design complexity for new compute systems for many decades now. However, the design of specialized accelerators also requires understanding of the underlying algorithm being targeted and the mathematical opportunities for algorithm re-formulation to accommodate constraints in the software and hardware design.* ***New co-design methodologies that span the full compute stack are required that fully integrate applied mathematics and algorithms into the co-design process together with software and hardware design. This will also require new kinds of Electronic Design Optimization tools to facilitate that process. Lastly, funding agencies will need to co-fund system development projects to incorporate these multidisciplinary co-design elements from the start, and not fund software,***

hardware, and algorithm development as separate un-integrated tasks.

- **New programming abstractions are needed.** *Existing programming abstractions were created with a universal von-Neumann architecture baked at a very fundamental level. With the emergence of non-Von-Neumann heterogeneous accelerators, the fundamentals of how to express an algorithm in a computer language will be upended.* **There is a significant need to explore new programming paradigms and programming languages that are able to express data locality and also target many different kinds of potential accelerators.**

- **Cross-disciplinary funding and research efforts are needed.** *Funding and research efforts are currently too siloed in particular disciplines. Successfully designing and implementing a compute system composed of different types of accelerators will require cross-disciplinary efforts. However, many funding agencies are split discipline-specific programs, which leads to single discipline-focused research.* ***Funding agencies will need to focus on funding interdisciplinary teams, and research efforts should focus particularly on challenges that span multiple levels of the compute stack.***

- **Generalists are needed to enable successful co-design.** *Our education systems have focused on training students to focus on particular disciplines; for example, many computer science students are even focusing on a particular subset of computer science, such as machine learning, and are missing the greater context of where and how their work fits into the broader field.* ***Education programs should be developed that focus on training generalists, who understand how to work across disciplines and can enable interdisciplinary collaborations.***

Many of our findings are treated as afterthoughts in today's accelerator design process. Compatibility and benchmarks are neglected as developers prioritize performance of accelerators as standalone entities. This, however, is not an efficient or realistic approach to accelerator design, as accelerators rely upon the entire compute stack.

Whereas our fundamental approach to separation of concerns separates each layer of the compute stack (from hardware to software to algorithms) into many largely independent and separately funded specialized activities, development for future systems that have potentially many different kinds of hardware acceleration require that we break-through those layers and do truly multidisciplinary co-design that spans those layers. This not-only changes how we approach the systems design process, but also will fundamentally change the way that research and development is funded for these future systems.

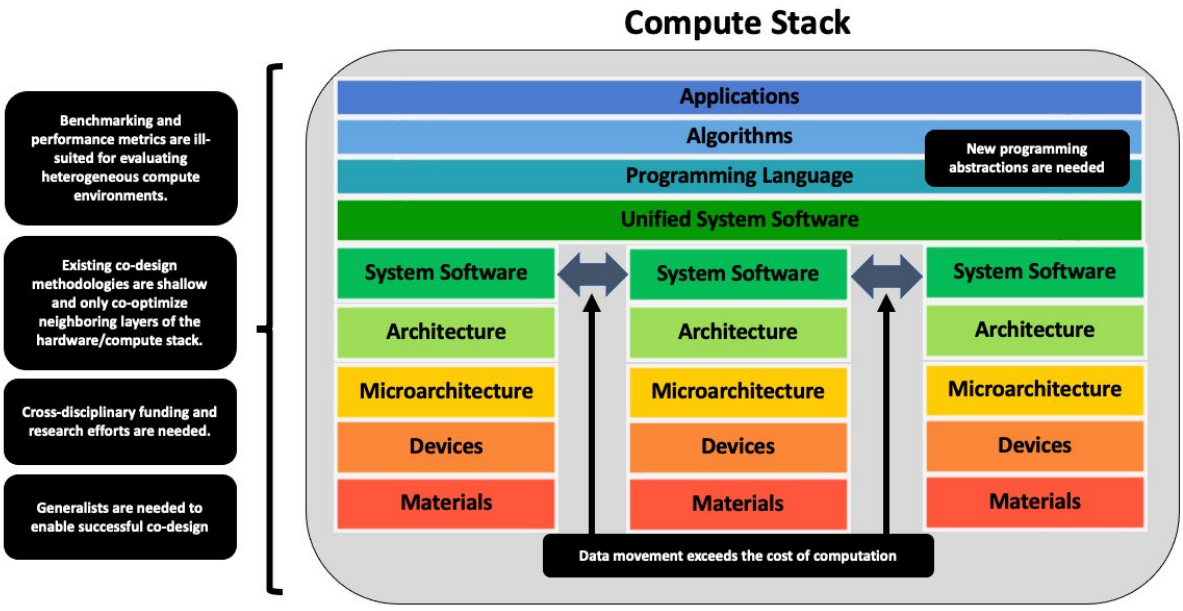

Figure 5: *The future of heterogeneous computing requires monumental restructuring in the development process, performance evaluation, and funding. (Catherine Schuman)*

MAJOR FINDINGS

Section 1: Programmability:

- **1.1 Algorithm/Software/Hardware Co-design:** Current research for systems is stratified where algorithms, software, and systems are funded on different tracks. Each of these systems are complex, and require a high level of expertise to develop, which makes it difficult to both find researchers with experience and knowledge in more than one area, and to facilitate conversations between these areas of development. However, there is a growing recognition of the need to integrate these components seamlessly in order to realize optimal performance and efficiency in computing systems. This entails a paradigm shift towards co-design methodologies that emphasize concurrent and collaborative development of algorithms, software, and hardware components. Such an approach, often likened to "co-design on steroids", underscores the importance of designing these components in tandem to unlock the full potential of specialization in future computing systems.

- **1.2 Data-locality and Data-Movement-Centric Programming:** Existing programming models are centered around specification of the computational operations in an algorithm, but ignore data locality and data movement. However, data movement is now more expensive than the computational operations. The memory hierarchy (including "levels", locations, proximity, consistency, and caching) poses a number of research questions. Research in academia and industry needs to explore how to specify "location" or "locality" of data, but also for the programmer to be able to reason explicitly about the costs of the data movement in the same way that "algorithmic order of complexity" is currently used to reason about the computational costs of a given algorithm. (example the Programming Abstractions for Data Locality international workshops series)

- **1.3 Standardized accelerator interfaces:** Currently, in accelerator development, every accelerator is equipped with its own complex set of low level software interfaces, commonly referred to as drivers, in order to access and utilize the capabilities of the accelerator. However, this approach incurs notable costs in terms of programmability and maintainability. The proliferation of diverse accelerators further exacerbates this issue, leading to an unsustainable surge in software complexity. With each new type of accelerator that emerges, the burden on developers and those that maintain the software grows, hindering scalability. It also erects barriers to seamless integration and interoperability of various accelerator technologies within the broader computing ecosystem. As a result, there is a pressing need for novel techniques that streamline the software interface landscape, foster greater standardization, and promote interoperability across diverse accelerator platforms.

Section 2: Accelerator Design:

- **2.1 Co-design methodology research:** Accelerators should not be developed in isolation. Instead, accelerator designers should consider the broader computing environment that accelerators will be integrated into. Such an ecosystem entails not only the hardware components of the accelerators themselves, and the hardware of the systems into which they are integrated, but also the intricate interplay with software frameworks, networking infrastructure, and data processing methodologies. Another crucial component of these systems which is often overlooked is people. The human component of development plays a pivotal role in the success and efficacy of accelerator technologies. Acknowledging the diverse needs, preferences, and skill sets of developers, end users, and other stakeholders is paramount. By considering the human dimension alongside technological considerations, accelerator designers can cultivate solutions that are not only technologically proficient but also user-friendly, intuitive, and aligned with real-world operational requirements. Additionally, defining success metrics for co-design is critical. As noted in Section 3, benchmarks and metrics are important to drive successful innovation, but it is not immediately clear how to measure the success of co-design approaches. New methodologies and guidelines for effective co-design should be developed, and metrics should be defined to gauge whether co-design efforts are successful.

- **2.2 Better programming abstractions:** Related to programmability and usability, better programming abstractions need to be defined alongside hardware. The current development landscape underscores the diverse array of accelerators, each necessitating distinct and often radically different approaches to hardware programming. Be-

cause of this, the traditional "one-size-fits-all" programming paradigms fall short of adequately addressing the nuanced requirements and intricacies of disparate accelerator technologies. Consequently, there is a compelling need to define and implement programming abstractions that accommodate the heterogeneous nature of accelerators and empower developers with intuitive, flexible, and efficient tools for harnessing the full potential of these hardware devices. Different accelerators require radically different ways of thinking about programming the hardware.

- **2.3 Early considerations of accelerator integration:** In the early stages of accelerator design, deliberate consideration of how the accelerator will be integrated in existing and future systems is necessary to ensure overall success and functionality. Many accelerators that are currently being developed or researched include non-CMOS devices; many of these devices are also not easily physically integrated with CMOS systems. It is critical that integration with existing technologies be considered early in the development of these new accelerators. Additionally, the physical placement and spatial requirements of suitable systems must be considered early on. Other questions, such as the degree of integration with traditional compute resources and the potential interactions with other accelerators need to be addressed proactively. Moreover, communication pathways and interfaces between the accelerator and other components must be meticulously defined to ensure interoperability and efficient data exchange. For example, accelerators may include analog components that will require analog-digital converters and digital-analog converters, which may add significantly to communication costs and should be considered as part of the design of the greater system. By incorporating integration considerations early in the design phase, developers can lay the foundation for a cohesive and optimized computing environment that maximizes the potential of accelerators while minimizing bottlenecks and compatibility issues.

Section 3: Metrics/Benchmarks:

- **3.1 Improve accelerator system benchmarking:** The benchmarking process must be modeled after "full system" benchmarking efforts, exemplified by frameworks like the Transaction Processing Performance Council benchmarks. Benchmarks in this system are carefully crafted to outline a comprehensive set of requirements for an auditable system, providing a standardized reference point for performance evaluation. A similar strategy should be employed for accelerator benchmarking, in which benchmark specifications delineate specific criteria and functionalities that are expected from a given accelerator within a broader computing environment. To do so requires defining the performance metrics you wish to evaluate, and also outlining the requisite hardware and software configurations, data sets, and operational parameters necessary to conduct meaningful assessments. Additionally, the benchmarking process should include developing a reference design that serves as a standardized implementation which meets all benchmarking requirements. Doing so will facilitate reproducibility and comparability across different evaluations and it will provide a tangible blueprint for stakeholders to assess the efficacy and performance of accelerator technologies in real-world scenarios. Adopting a holistic and standardized approach to benchmarking will allow researchers to foster transparency and accountability in evaluating accelerator performance.

- **3.2 Comprehensive metrics and figures of merit:** It is imperative that we adopt a comprehensive array of metrics that extend beyond mere temporal acceleration in evaluating the efficacy and viability of accelerator technologies. While speed remains a crucial aspect, a holistic assessment framework must incorporate a diverse criteria of metrics spanning energy consumption, power efficiency, circular life cycle sustainability, SWaP (size, weight and power), security across the supply chain and operation phases, privacy considerations, and productivity enhancement. However, we must acknowledge that quantifying and evaluating many of these metrics pose significant challenges, as they often entail complex and multifaceted interactions within the computing ecosystem. Energy consumption and power efficiency, for instance, require measurement methodologies that capture direct power usage and ancillary factors such as cooling systems. Circular life cycle sustainability requires assessing the environmental impact of materials sourcing, production processes, and end-of-life disposal or recycling mechanisms. Similarly, ensuring security across the supply chain and operational life cycle involves comprehensive risk assessments and

robust mitigation strategies. Privacy considerations necessitate adherence to stringent data protection protocols and regulatory compliance frameworks. Finally, enhancing productivity entails not only optimizing computational performance but also streamlining development workflows and facilitating seamless integration with existing software ecosystems. Though integration of each of these diverse metrics in an overall assessment framework is a monumental challenge, it would allow a much more comprehensive evaluation of these technologies, and would reduce waste and inefficiency. While a single figure of merit encourages competition, we believe that multiple rankings should be maintained, such as those based on raw time performance, on time performance per watt, etc., ratioed to the reference design.

Section 4: Education/Funding Ecosystem:

- **4.1 Computer science education is overly compartmentalized:** The current state of education in computing suffers from compartmentalization. Students are not required to learn about each level of the computational stack, and many graduate with significant gaps in their knowledge of the full stack. Interdependencies, however, are becoming increasingly pronounced across the compute stack as more diverse accelerators are incorporated. Future computing education requires a curriculum that fosters a comprehensive understanding of the entire computational stack, from hardware design and system architecture to software development and optimization.

- **4.2 Funding for computing research is siloed:** At present, funding for disciplines such as computing and hardware is siloed, with set amounts of funds allocated towards one or the other, but little to no funding is set aside for interdisciplinary projects. This further exacerbates the issue above, of students failing to adequately learn about each area of the computational stack. The way we teach computing needs drastic remodels, down to even introductory courses.

- **4.3 The barrier to entry for accelerator development is too high:** There is currently too high of a barrier for accelerator development and integration (in terms of costs). Accelerators require substantial investment in research, development, prototyping, and testing. These costs include both the procurement of specialized hardware components and the recruitment of skilled personnel. Furthermore, the complexity of integrating accelerators into existing computing infrastructures adds to the financial burden, as it often requires modifications to hardware architectures, software frameworks, and data workflows. Regulatory compliance, intellectual property considerations, and market competition further compound costs associated with accelerator development and integration. Consequently, this high barrier to entry poses challenges for startups, research institutions, and even established companies.

Recommendations

Section 1: Programmability

- **1.1 Deep algorithm/software/hardware co-design:** Programming environments, algorithms (applied math) and software must be funded contemporaneously with the hardware/system development process (co-design on steroids) rather than the current model that funds these as separate R&D tracks. This recommendation pertains to how government funding agencies should fund R&D projects by creating these multi-faceted multidisciplinary teams to address grand challenges of computing rather than funding these efforts separately.

- **1.2 Data-centric programming systems and algorithms:** As data movement is more costly in terms of power and performance than the computation in many cases. Moving to a data-centric approach to programming models and systems will be crucial to expressing and exposing those costs. This paradigm is an inversion of existing deep-rooted compute-centric programming paradigms that are focused primarily on economizing on the computation. Even computational complexity (as opposed to data movement complexity) is deeply rooted in academic curriculums. Addressing these challenges requires re-evaluation of the foundations of computational science.

- **1.3 Accelerator interface standards:** Development of standardized/reusable (driverless) programming interfaces would eliminate many of the most important barriers to programmability and maintainability for future heteroge-

neous acceleration. This has parallels to the importance that MPI played in normalizing the differences between the underlying networking hardware for early HPC clusters. This must be done on a pre-competitive basis so that solutions could be deployed in a non-proprietary/platform-agnostic manner. This will require tight integration of academic and industry efforts funded by the government, or alternatively the development of an industry consortium similar to standards bodies like JEDEC.

Section 2: Education

- **2.1 In education, we need efforts across the pipeline (from K12 through continued workforce training):** Computing education must span the entire continuum of the education pipeline, from K-12 schooling all the way through continued workforce training and professional development initiatives. Beginning with foundational education in primary and secondary schools is crucial to ensure that youth consider the wide field of computing as a professional career early on. Students should be instructed to consider computing beyond its applications to other careers and should be taught the basics behind standard computing systems (hardware, software, and ethical considerations with these systems). Moreover, in bridging the gap between academic learning and practical application, more vocational training programs and apprenticeships should be offered and clearly communicated with young researchers to develop and foster their interest in computing research.

- **2.2 We need to focus on creating/educating generalists in addition to specialists:** There is a growing recognition of the importance of fostering generalists who possess a broad range of skills and knowledge across multiple computing disciplines. Generalists are individuals who can adapt to diverse environments, integrate knowledge from different domains, and tackle complex problems from a holistic perspective. More of these individuals are needed to communicate the needs of different levels of development across the compute stack for specific systems. Often during development, each hardware and software component of a given system is developed contemporaneously, however these developer groups rarely communicate the requirements and goals of their specific components until the integration stage, long after their components have been outlined. This leads to inefficient software and hardware components that are not maximized. Generalists can communicate across development groups to ensure that new hardware is integrated in a way that complements existing hardware. Similarly, software can be written with an understanding of what hardware components require, in what order they access data and compute power, and can minimize inefficiencies and duplicative efforts.

Section 3: Funding

- **3.1 Lowering the barrier of entry in accelerator development and integration is essential for fostering innovation and broadening participation to advance computing technologies:** Presently, the cost of entry into accelerator development is prohibitively high for many individuals and organizations, limiting access to resources and opportunities in this field to organizations with huge amounts of resources, financial and otherwise. To address this challenge, funding agencies must take proactive measures to support initiatives that reduce costs, provide access to affordable hardware and software tools, and offer training and educational resources to aspiring accelerator developers. Funding agencies could also connect small organizations and startups who, separately, do not have the resources necessary to break into the field of accelerator development. Investing in programs that promote inclusivity and diversity in accelerator development would grant more individuals the opportunity to enter the field, and in turn may promote unique ideas that may not be pursued by larger companies, who are often motivated by profit over exploration. A more inclusive funding strategy, beyond benefiting just the developers and organizations involved, would also benefit the wider society by broadening the landscape of accelerators being developed.

- **3.2 Increase funding mechanisms for interdisciplinary co-design:** There need to be more mechanisms for funding large, interdisciplinary teams for co-design. Advancing accelerator co-design requires the collaboration of large, interdisciplinary teams with expertise in diverse fields, such as computer science, electrical engineering, mathematics, and materials science. However, existing funding mechanisms are usually tailored to supporting

research conducted by experts in one discipline. These funding mechanisms lead to research that is limited in scope, and often only addresses one aspect of a computing system. To address the full potential of accelerator co-design, more flexible and comprehensive funding mechanisms must be established.

- **3.3 Interdisciplinary collaboration needs to be thought of and funded differently than traditional research programs:** These mechanisms should be designed to provide resources and support, curated for groups of researchers with a wide range of expertise. Interdisciplinary groups require more guidance, such as defining terminology to limit misunderstandings and planning more meetings to keep each member up to date on what another may be working on. Funding mechanisms must allocate resources towards identifying and choosing a group leader, with knowledge of each of the research domains that make up the group. This leader must be a skilled communicator, a generalist, who can understand the needs of each subgroup on a project, and explain these needs to the whole group and the funding institution(s). Interdisciplinary collaboration is more difficult to facilitate, but it is not realistic to pursue many worthwhile research projects from a single disciplinary standpoint.

Section 4: Accelerator design

- **4.1 Co-design methodology and tools:** Development of co-design methodology and co-design tools for accelerator design is critical, and funding efforts should focus on interdisciplinary groups to facilitate this development.

- **4.2 Programming abstractions need to be developed alongside hardware, not independently of it:** By developing programming abstractions in conjunction with hardware, developers can leverage the capabilities and features of the hardware more effectively, leading to more efficient code execution and better utilization of resources. Moreover, this approach fosters synergy between hardware and software development efforts, facilitating the creation of integrated solutions that are well-suited to the specific requirements and characteristics of the underlying hardware platform. Ultimately, aligning programming abstractions with hardware advances results in more robust and scalable software systems.

- **4.3 Consensus on the best practices for evaluating accelerator systems must be reached:** In order to do so, the community of accelerator developers and researchers needs to come together and form a consortium to shepherd development of new benchmarks and effective, meaningful metrics.

Workshop participants

First Name	Last Name	Affiliation
Alex	Aiken	Stanford University
Brad	Aimone	Sandia National Laboratories
Shelah	Ameli	The University of Tennessee, Knoxville
Katerina	Antypas	Lawrence Berkeley National Laboratory
Hartwig	Anzt	University of Tennessee
Randal	Burns	Johns Hopkins University
Kirk	Bresniker	Hewlett Packard Enterprise
Kirk	Cameron	Virginia Tech
Tracy	Camp	Computing Research Association
Tom	Conte	Georgia Institute of Technology
Anshu	Dubey	Argonne National Laboratory
Farzad	Fatollahi-Fard	Lawrence Berkeley National Lab
Franz	Franchetti	Carnegie Mellon University
Hubertus	Franke	IBM Research
Cat	Gill	Computing Community Consortium
Hector	Gonzalez	SpiNNcloud Systems GmbH
Haley	Griffin	Computing Community Consortium
Madeline	Hunter	Computing Community Consortium
Neena	Imam	NVIDIA
Anirudh	Jain	Georgia Tech
Robin	Knauerhase	AMD Advanced R&D
Piotr	Luszczek	University of Tennessee
Edwin	Ng	NTT Research
Vivek	Sarkar	Georgia Institute of Technology
Catherine	Schuman	University of Tennessee
John	Shalf	Lawrence Berkeley National Laboratory
Michelle	Strout	Hewlett Packard Enterprise
Neil	Thompson	Massachusetts Institute of Technology

CCC

Computing Community Consortium
Catalyst

1828 L Street, NW, Suite 800
Washington, DC 20036
P: 202 234 2111 F: 202 667 1066
www.cra.org cccinfo@cra.org