Emerging Heterogeneous Systems Provide Great Opportunities for Codesign

Aaron R. Young, Jeffrey S. Vetter, Frank Liu,
Narasinga Rao Miniskar, Sarat Sreepathi, and Anthony M. Cabrera
Oak Ridge National Laboratory, Oak Ridge, TN, USA
{youngar,vetter,liufy,miniskarnr,sarat,cabreraam}@ornl.gov

Topic: Architectures, Applications, Modeling and Simulation, Emerging Technologies

1 Challenge

As Moore's Law and Dennard scaling are coming to an end, simple technology scaling cannot be relied on for performance gain, and new technologies and computing paradigms must be developed to continue improving application performance. To this end, hardware accelerators like GPUs, TPUs, and FPGAs are being employed as co-processors to traditional systems to accelerate computation. Additionally, new ways of computing, for example, neuromorphic and quantum computing, are also showing promise and could be incorporated into large-scale HPC systems. Because different scientific domains and applications will benefit from different configurations of accelerator types and computing paradigms, we predict that HPC systems will become "extremely" heterogeneous [1]. Though such systems could significantly accelerate application performance, there are many resulting application, system, and hardware development challenges, including: 1) how to write software to target extremely heterogeneous systems, 2) how to port legacy software to new systems, 3) how the programming environment and runtime system should map the software to utilize the hardware accelerators efficiently, and 4) how the heterogeneous systems should be designed (i.e., which accelerators should be included and how they should be configured).

We believe that by codesigning the applications, runtime environment, and accelerators, it is possible to achieve high-performance without losing performance portability in future, extremely heterogeneous systems.

2 Opportunity

Interconnect The challenge of how to design these heterogeneous systems leads to many research opportunities. One such opportunity is architecture exploration of the interconnect between compute and memory components. CXL, an emerging and open industry standard processor interconnect, is designed to enable low-latency memory access and create coherent memory space between CPUs, accelerators, and memory pools. CXL is a promising interconnect to enable multiple accelerators and new emerging memory technologies to be incorporated within an extremely heterogeneous node.

Architecture Modeling The interconnect, applications, runtime, and accelerators all need to be designed together to meet performance and compatibility requirements. Architecture level models built using tools like GEM5 and SST will enable the joint exploration of all these components using simulation. Architecture level simulations and machine learning techniques can be used to perform design space exploration to determine the best architecture configuration [2].

Hardware Design Accelerators are more specialized than general compute cores. Some devices like FPGAs are re-configurable, while others perform more limited operations. In all cases, a joint design effort by hardware and software designers is required to develop software that can leverage the more specialized hardware and design hardware that is well suited to accelerate important software tasks.

Application Characterization Workload characterization of key target application(s) is pivotal to a successful co-design effort in informing design space exploration. An in-depth analysis of

applications and associated proxy apps [3] would entail both static and dynamic analyses as well as gathering architecture-aware and agnostic metrics. The proliferation of high-bandwidth memory and complex memory hierarchies necessitate research into trade-offs for memory-bound applications.

Heterogeneous Programming How to best write maintainable code that can target a range of different accelerators is an ongoing challenge. However, current work is looking at the portability of popular accelerator languages such as OpenCL and C. Additionally, new frameworks like OneAPI, SYCL, and OpenARC [4] show promise for enabling heterogeneous compute; however, additional work is needed on both the language and hardware sides to ensure the performance portability of software written in these frameworks.

Runtime Another considerable challenge and opportunity is in designing runtime frameworks to map the work expressed by the application onto hardware for execution. MPI and OpenMP are widely used solutions for expressing concurrency for distributed applications, but these methods will fall short when targeting large heterogeneous systems. More complex heterogeneous workflows could be expressed using a task-based or dataflow-based programming model. Work expressed in these models could then be automatically parallelized and distributed to the appropriate compute nodes and accelerators by a runtime system. This system could leverage multiple tuned application kernels that were either provided or generated from a higher-level language by a compiler. The runtime could also use architecture-independent workload characteristics [5] and workload-independent hardware characteristics, along with an AI-assisted scheduling policy to assign tasks and map the work across the heterogeneous HPC system.

3 Timeliness

The end of simple compute scaling, the rise of open hardware, and the increase in new accelerators are leading to increasingly heterogeneous architectures. New developments in heterogeneous languages and runtimes are enabling the software support to leverage this heterogeneous hardware. By codesigning the applications, programming languages, runtimes, system architectures, and accelerator hardware, future heterogeneous HPC systems are designed to accelerate workloads beyond what is currently possible. If we work to codesign these systems now, we can overcome these challenges and create maintainable, performance portable code.

References

[1] Jeffrey S Vetter, Ron Brightwell, Maya Gokhale, et al. *Extreme heterogeneity 2018-productive computational science in the era of extreme heterogeneity: Report for DOE ASCR workshop on extreme heterogeneity.* Tech. rep. USDOE Office of Science (SC), Washington, DC (United States), 2018.

[2] Frank Liu, Narasinga Rao Miniskar, Dwaipayan Chakraborty, et al. "Deffe: A Data-Efficient Framework for Performance Characterization in Domain-Specific Computing". In: *Proceedings of the 17th ACM International Conference on Computing Frontiers.* CF '20. Catania, Sicily, Italy: Association for Computing Machinery, 2020, pp. 182–191. ISBN: 9781450379564. DOI: 10.1145/3387902.3392633.

[3] Sarat Sreepathi, M. L. Grodowitz, Robert Lim, et al. "Application Characterization Using Oxbow Toolkit and PADS Infrastructure". In: *Proceedings of the 1st International Workshop on Hardware-Software Co-Design for High Performance Computing.* Co-HPC '14. New Orleans, Louisiana: IEEE Press, 2014, pp. 55–63. ISBN: 9781479975648. DOI: 10.1109/Co-HPC.2014.11. URL: https://doi.org/10.1109/Co-HPC.2014.11.

[4] S. Lee, D. Li, and J. S. Vetter. "Interactive Program Debugging and Optimization for Directive-Based, Efficient GPU Computing". In: *2014 IEEE 28th International Parallel and Distributed Processing Symposium.* May 2014, pp. 481–490. DOI: 10.1109/IPDPS.2014.57.

[5] B. Johnston and J. Milthorpe. "AIWC: OpenCL-Based Architecture-Independent Workload Characterization". In: *2018 IEEE/ACM 5th Workshop on the LLVM Compiler Infrastructure in HPC (LLVM-HPC).* Nov. 2018, pp. 81–91. DOI: 10.1109/LLVM-HPC.2018.8639381.