Extreme Heterogeneity with Resilience by Design (and not as an Afterthought)[1]

Christian Engelmann, Rizwan Ashraf, and Saurabh Hukerikar
Oak Ridge National Laboratory, Oak Ridge, TN, USA

Key Challenges: Resilience, i.e., obtaining a correct solution in a timely and efficient manner, is one of the key challenges in extreme-scale high-performance computing (HPC). Extreme heterogeneity, i.e., using multiple, and potentially configurable, types of processors, accelerators and memory/storage in a single computing platform, will add a significant amount of complexity to the HPC hardware/software ecosystem. The notion of correct computation and program state assumed by users and application developers today, which has been based on binary bit-level correctness, will no longer hold for processing elements based on quantum qubits and analog circuits that model spiking neurons in neuromorphic computing elements. The diverse set of compute and memory components in future heterogeneous systems will require novel hardware and software resilience solutions. Errors and failures reported by such heterogeneous hardware will need to be handled by the appropriate software component to enable efficient masking, recovery, and avoidance with little burden on the user. Similarly, errors and failures reported by the software running on such heterogeneous hardware need to be equally efficiently handled with little burden on the user. This requires a new approach, where resilience is holistically provided by the HPC hardware/software ecosystem. The key challenges are to design and to operate extreme heterogeneous HPC systems with (1) wide-ranging resilience capabilities in system software, programming models, libraries, and applications, (2) interfaces and mechanisms for coordinating resilience capabilities across diverse hardware and software components, (3) appropriate metrics and tools for assessing performance, resilience, and energy, and (4) an understanding of the performance, resilience and energy trade-off that eventually results in well-informed HPC system design choices and runtime decisions.

Research Direction: Coordinated cross-layer and adaptive resilience solutions can offer efficient error and failure masking, recovery, and avoidance at the appropriate hardware or software component and compute or data granularity. While the various heterogeneous compute and memory components will likely have hardware resilience mechanisms, software-based solutions to fill gaps in detection, masking, recovery, and avoidance of errors and failures will require coordination between the multiple layers of the system. Based on the underlying execution model and intrinsic resilience features of the hardware, the various components in an extreme heterogeneous system may be organized into predefined protection domains. Coordinated resilience solutions will handle errors and failures in specific components and granularities where it is most appropriate to do so and in coordination with the rest of the system, which prevents errors from propagating and failures from cascading beyond the protection domains. This approach also removes some of the complexity that is introduced by extreme heterogeneity. Adaptive strategies can leverage the unique capabilities of the heterogeneous protection domains, since the performance, resilience and energy profiles of each domain are different. Programming models and runtime environments may dynamically configure an application to use specific components in the heterogeneous system based on performance, resilience and energy costs. For example, critical computation may be executed on more resilient components; computation on less resilient components may be checked for errors with computation on more resilient components. Critical data may be stored solely or at least backed up on more resilient storage. Holistic cross-layer and adaptive resilience essentially provides efficient end-to-end resilience for computation and data. Future research needs to address the following aspects:

- Enabling wide-ranging resilience capabilities in system software, programming models, libraries, and applications requires identifying error and failure models, protection domains, programming interfaces, and implementation mechanisms. Resilience needs to be an integral part of the HPC hardware/software ecosystem, such that the burden for it is on the system by design and not on the user as an afterthought.
- Coordinating resilience across hardware and software components, either statically with policies or dynamically using a runtime, requires interfaces and mechanisms for communicating resilience capabilities and quality of service requirements, and for configuring the capabilities to meet the requirements.

[1] This work was sponsored by the U.S. Department of Energy's Office of Advanced Scientific Computing Research. This manuscript has been authored by UT-Battelle, LLC under Contract No. DE-AC05-00OR22725 with the U.S. Department of Energy.

- Uniform metrics for assessing performance, resilience and energy across heterogeneous components are required to enable design and run time trade-offs. Design space exploration tools are needed to make trade-off decisions at design time for static cross-layer and adaptive resilience, while monitoring tools and runtime environments are needed to make trade-off decisions for dynamic cross-layer and adaptive resilience.
- Understanding the performance, resilience and energy trade-off is key to solving the resilience challenge for extreme heterogeneity, which is to design and operate a reliable system within a given cost budget to achieve an expected performance. Design choices and runtime decisions are based on a detailed understanding of the trade-off, which is HPC system and HPC application specific.

In addition, future efforts also need to communicate best practices and standards for resilience in HPC.

State of the Art: The state of practice for HPC resilience is global application-level checkpoint/restart, a single-layer approach that burdens the user with employing a resilience strategy at extreme coarse granularity (the job level). Part of the state of practice for HPC resilience are also hardware solutions at extreme fine granularity, such as SECDED ECC for main memory, caches, registers and architectural state, Chipkill for main memory, redundant power supplies and voltage regulators, and RAS management systems for monitoring and control. The state of research is more advanced and includes a number of resilience technologies, such as fault-tolerant MPI, redundant MPI, proactive fault tolerance, containment domains, and resilient solvers. Work in understanding the performance/energy and performance/resilience trade-offs exists as well. Recent work [1] made inroads in understanding the fault, error and failure models of HPC systems. Other recent work [2] pioneered the concept of design patterns for a structured approach to HPC resilience and employs this concept for cross-layer and adaptive resilience in a research prototype.

Assessment

Maturity: The state of research for HPC resilience is rich in mechanisms that can be utilized. However, a longer-term and coordinated effort is required to enable wide-ranging resilience capabilities in practice and to make them an integral part of the HPC hardware/software ecosystem. This includes creating best practices and standards for resilience in operating and runtime systems, the programming models, numerical libraries, workflow engines, and applications. The recent work [1,2] in resilience design patterns and in cross-layer and adaptive resilience has pioneered some of the core technology for coordinating resilience through design. However, research in defining, communicating and matching HPC resilience capabilities and quality of service requirements is required as we transition to extreme heterogeneity. The recent work in fault models and in various trade-offs can form the basis for the metrics and trade-off challenges. However, more research in uniform metrics, in performance, resilience and energy trade-offs, in design space exploration tools, and in adaptive runtimes is still required.

Timeliness: Simply put, if resilience by design is not done now, in the early stages of extreme heterogeneity, the current state of practice for HPC resilience, global application-level checkpoint/restart, will remain the same for decades to come due to the high costs of adoption later on. The prime example for this is MPI, for which, 25 years after its first standardization, resilience extensions are still not part of the MPI standard. In contrast to the MPI standardization effort 25 years ago, the current state of research for HPC resilience is far beyond the current state of practice. The existing knowledge, experience and prototypes serve as a foundation for making resilience an integral part of the HPC hardware/software ecosystem.

Uniqueness: Resilience is a challenge that is unique to scientific applications of supercomputing due to cost constraints that limit resilience mitigation in hardware by vendors. This requires a cooperative approach between HPC hardware and software to efficiently mitigate faults, errors, and failures.

Novelty: The proposed research targets making resilience an integral part of the HPC hardware/software ecosystem by design, which requires participation by the respective owners of the HPC hardware/software ecosystem. While interaction with vendors is needed for innovation on the HPC hardware side, the planned work on the HPC software side in operating and runtime systems, programming models, numerical libraries, workflow engines, and applications should be done through the ASCR program that owns these to assure an early and low cost adoption and to assure the targeted low burden on the user for HPC resilience.

References

[1] S. Gupta, T. Patel, C. Engelmann, and D. Tiwari. “Failures in Large Scale Systems: Long-term Measurement, Analysis, and Implications,” SC 2017, Denver, CO, USA, November 12-17, 2017.

[2] S. Hukerikar and C. Engelmann. “Resilience Design Patterns: A Structured Approach to Resilience at Extreme Scale,” Journal of Supercomputing Frontiers and Innovations (JSFI), 4(3):4-42, 2017.