Resilience by Codesign (and not as an Afterthought)

Christian Engelmann, Ph.D.

Senior Scientist & Group Leader
Intelligent Systems and Facilities Group
Advanced Computing Systems Research Section
Computer Science and Mathematics Division
Oak Ridge National Laboratory

Motivation

- Resilience: Obtaining a correct solution in a timely and efficient manner
- Resilience is a major challenge in future HPC systems
  - Always first of its kind in production (brand new hardware and/or software)
  - Increased component counts (18,688 GPUs in Titan → 27,648 in Summit)
  - Decreased individual component reliability (e.g., due to process technology issues)
  - Increased software complexity (e.g., due to scale and diversity)
- Additional concerns emerge with extreme heterogeneity
  - More and different accelerators (Reliability? Interdependencies?)
  - Complex memory hierarchies (Reliability? Interdependencies?)
  - Non von Neumann accelerators (Reliability? What is correctness?)

Current State of Practice

- Global application-level checkpoint/restart
  - Extreme coarse grain solution
  - Burdens the user with employing the resilience strategy
- Hardware solutions exist at extreme fine granularity
  - SECDED ECC for main memory, caches, registers and architectural state
  - Chipkill for main memory
  - Redundant power supplies and voltage regulators
  - …
- RAS management systems for monitoring and control

Current State of Research

- Fault-tolerant programming models
  - Fault-tolerant MPI, re-execution of failed tasks and containment domains
- Proactive fault tolerance using migration of computation away from components that are about to fail
- Resilient solvers with recovery, compensation or self-stabilization
- Understanding the fault, error and failure characteristics of HPC systems
- Understanding the performance/energy and performance/resilience trade-offs in HPC systems
- Design patterns for a structured approach to HPC resilience

Limitations of the Current State of Practice/Research

- Hardware/software HPC codesign for resilience is mostly nonexistent
- There are no design space exploration tools investigating the trade-offs
- As a result, HPC resilience research solutions are not adopted in practice
  - The cost/benefit trade-off of adoption is unknown
- Another result is the inability to mitigate reliability issues
  - There is a lack of alternatives in the production resilience "toolbox"
  - This results in degraded capability:
    - SC20 paper about ORNL Titan GPU failures
    - Recent Facebook Engineering paper on silent data corruption

We need Resilience by Codesign!

- Resilience needs to become an integral part of the HPC hardware/ software ecosystem through codesign
- The burden for resilience should be on the system by design and not on the operator or user as an afterthought
- Resilience in extreme-scale supercomputers is an optimization problem between the key design and deployment cost factors:
  - Performance, resilience, and power consumption
- The main challenge is to codesign a reliable system within a given cost budget that achieves the expected performance

Proposed Approach

- Codesign coordinated cross-layer and adaptive resilience solutions
  - Offer efficient error and failure masking, recovery, and avoidance at the appropriate hardware or software component and compute or data granularity
  - 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
- Key challenge is to codesign extreme heterogeneous HPC systems with
  - **Wide-ranging resilience capabilities** in architecture, system software, programming models, libraries, and applications
  - **Interfaces and mechanisms for coordinating** resilience capabilities across diverse hardware and software components
  - **Appropriate metrics and tools** for assessing resilience
  - An understanding of the **performance, resilience and energy trade-off** that eventually results in well-informed system design choices

Future Research and Development Areas

- **Develop an understanding** of the error and failure characteristics of hardware and software components
- **Identify protection domains, interfaces and mechanisms** of resilience capabilities in hard- and software
- **Design interfaces and mechanisms** for coordinating resilience capabilities and quality of service requirements across hardware and software components
- **Define uniform metrics** for assessing performance, resilience and energy across heterogeneous components to enable design trade-offs
- **Create design space exploration tools** to understand the performance, resilience and energy trade-offs between different node and system designs

Timeliness/Maturity

- The state of HPC resilience research is rich in solutions that can be utilized
- The HPC resilience research community has a lot of experience/expertise
- The existing knowledge, experience and prototypes serve as a foundation for making resilience an integral part of the HPC hardware/software ecosystem
- If resilience by codesign is not done now, then the current state of practice for HPC resilience will remain the same for decades to come
  - After 26 years, MPI is still not fault tolerant, while PVM was 28 years ago in 1993

Resilience by Codesign (and not as an Afterthought)

Christian Engelmann, Ph.D.

Senior Scientist & Group Leader
Intelligent Systems and Facilities Group
Advanced Computing Systems Research Section
Computer Science and Mathematics Division
Oak Ridge National Laboratory

ORNL is managed by UT-Battelle, LLC for the US Department of Energy