Using Global View Resilience (GVR) to add Resilience to Exascale Applications

Hajime Fujita\(^1\), Pan Nan\(^1\), Aiman Fang\(^1\), Zachary A. Rubenstein\(^1\), Ziming Zheng\(^1\), Kamil Iskra\(^1\), Jeff Hammond\(^1\), Anshu Dube\(^1\), Pavan Balaji\(^1\), Andrew A. Chien\(^1,2\)

\(^1\)University of Chicago \(^2\)Argonne National Laboratory


**Motivation**

- Current high performance systems have achieved 10^15 FLOPS and progress towards 10^18 FLOPS.
- The exascale systems are comprised of millions of components, leading to higher error rates. It is anticipated that the mean time between failures (MTBF) could be less than an hour [1].
- Resilience becomes a major concern.
- Need for a new tool which addresses resilience issues.

**Global View Resilience**

- A new library that exploits a global view data, and adds reliability to globally visible data [2, 3].
- **Key Features**
  - Multi-version, multi-stream, distributed array: preserves critical application data with fine-grain manner, enables parallel recovery from complex errors such as latency errors.
  - Open resilience: maximizes recoverable errors with cross-layer partnership, leverages application-level error handling with unified error handlers.
  - Portable, flexible, application-controlled resilience.
  - Demonstrated scalable resilience with gentle slope and flexible forward error recovery.
  - Implemented as a library, which can be used together with other libraries (e.g. MPI, Trilinos), allowing modular migration to existing applications, or as a backend of other libraries/programming models.

**Multi-version, Multi-stream, Distributed Arrays**

- Exploits a global-view data model, which enables resilient, adaptive algorithms and variable scalability.
- Provides an abstraction of data representation which offers resilience and seamless integration of various components of memory/storage hierarchy.

**Rich Application Studies**

- Molecular Dynamics: miniMD (SNI Manfred Project), ddMD (LINL).
- Data Analytics: miniVE (SNI Manfred Project), PCG, GMRES.
- Computation Library: Trilinos (SNI).
- Monte Carlo Transport: OpenMC (ANL, CESAR co-design center).

**Performance Study**

To measure the runtime overhead of GVR, experiments using OpenMC, ddMD, and Chombo were conducted. Experiments for OpenMC and ddMD were done on Midway, whereas the experiments for Chombo were conducted on NERSC Edison. As for the MPI library MVAPICH2-2.0 on Midway and Cray MPT 7.0.0 on Edison were used.

**Future Work**

- Efficient multi-version implementation, including efficient differences, compression, and efficient exploitation of NV-VIM.
- Work with community to establish Open Resilience APIs, infrastructure and portable error types/handling.
- Additional application studies, scalability.
- Efficient portability studies, varying underlying hardware.

**References**


**Acknowledgments**

We thank Mark Hoormann, Mike Henness, and Keta Taranishi for giving useful discussions on Trilinos and linear solvers, Brian van Straalen for advice on Chombo, Ignacio Laguna, David Richards for insights on ddMD; and John R. Traimm, Andrew R. Siegel for supports on OpenMC.