NC STATE UNIVERSITY

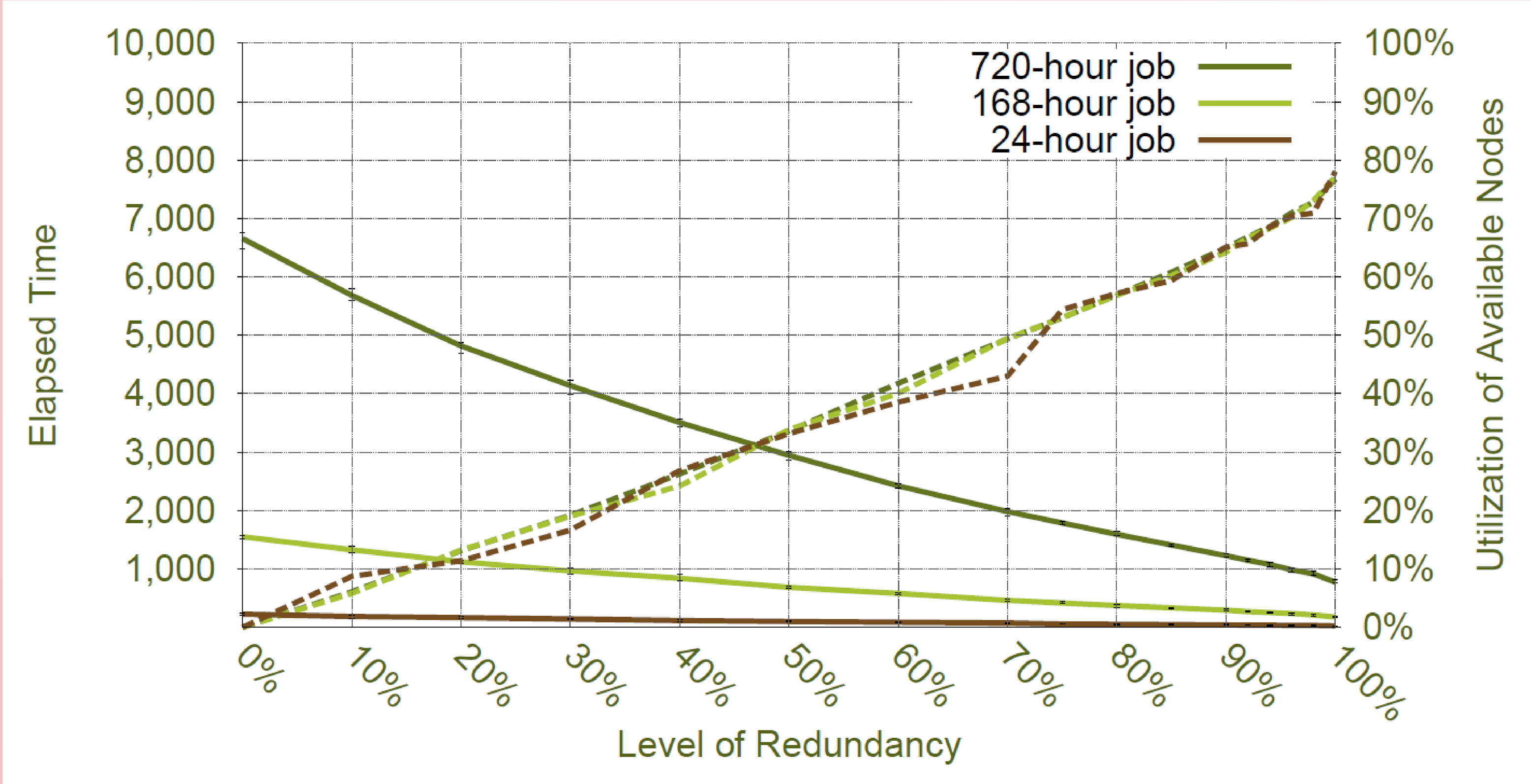
Detection and Correction of Silent Data Corruption for Large-Scale High-Performance Computing

David Fiala Advisor: Frank Mueller (NCSU)

Collaborators: Christian Engelmann (ORNL), Rolf Riesen, Kurt Ferreira (SNL)

Motivation

- Component failures require support of checkpoint/restart (C/R)
- Adding hardware increases the likelihood of faults
  - The probability of component failure combinatorially explodes
  - The mean-time-between-failure (MTBF) shortens
  - Overhead due to C/R increases exponentially
    - Computation vs. overhead ratio can be between 85%-55%
- Redundancy can reverse this trend
  - Each redundant process decreases the probability of failure of replica processes
  - Less interruptions produces greater utilization
  - 100% redundancy provides 5x job throughput [Sandia]



- Silent Data Corruption (SDC) faults manifest themselves as bit-flips in storage or even within processing cores
  - In some cases bit-flips are not correctable or even detected
  - Exacerbating this situation, when SDC goes undetected invalid results are reported
    - Memory becomes corrupt, but applications continue to run
    - This is a severe problem for today's large-scale simulations

Contributions

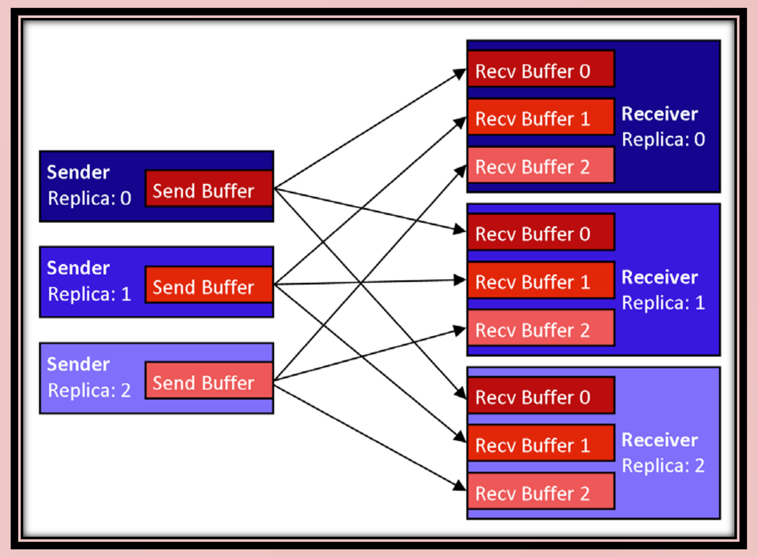
- Design and implementation of efficient mechanisms for fault tolerance in HPC
  - Propose efficient protocols for SDC protection
  - Investigate the cost of different levels of redundancy
- Demonstrate capabilities of SDC protection at the communication layer
  - Through fault injection we study failures in a native cluster environment

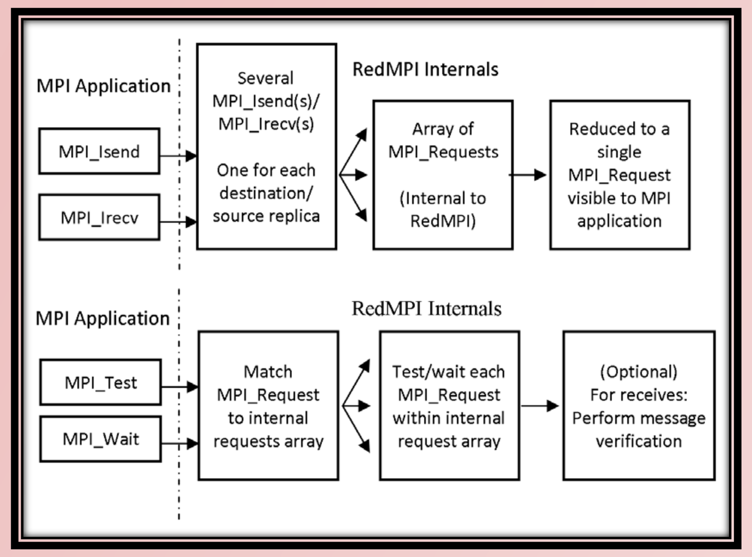
Design

- Provide transparency by linking unmodified MPI applications with our library: RedMPI
- RedMPI provides redundancy to MPI applications by instrumenting the MPI profiling layer
  - Adjusted MPI rank and size provide illusion of normal rank numbers
  - SDC protection is afforded by augmenting MPI_Isend, MPI_Irecv, and MPI_Wait/MPI_Test to communicate with replicas

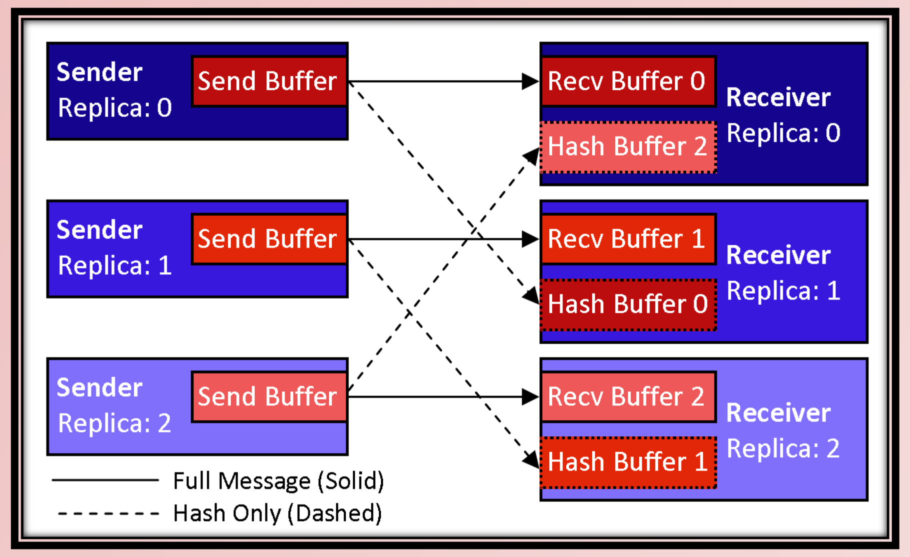
	No Redundancy	Dual Redundancy	Triple Redundancy or higher
Live SDC Detection	✗ No	✓ Yes	✓ Yes
Live SDC Correction	✗ No	✗ No	✓ Yes (via voting algorithm)

- Naïve SDC protection may be achieved by transmitting and comparing *r*r* messages amongst *r* total replicas.
  - Induces high interconnect contention / bandwidth degradation
  - Compare received buffers, discard a mismatch



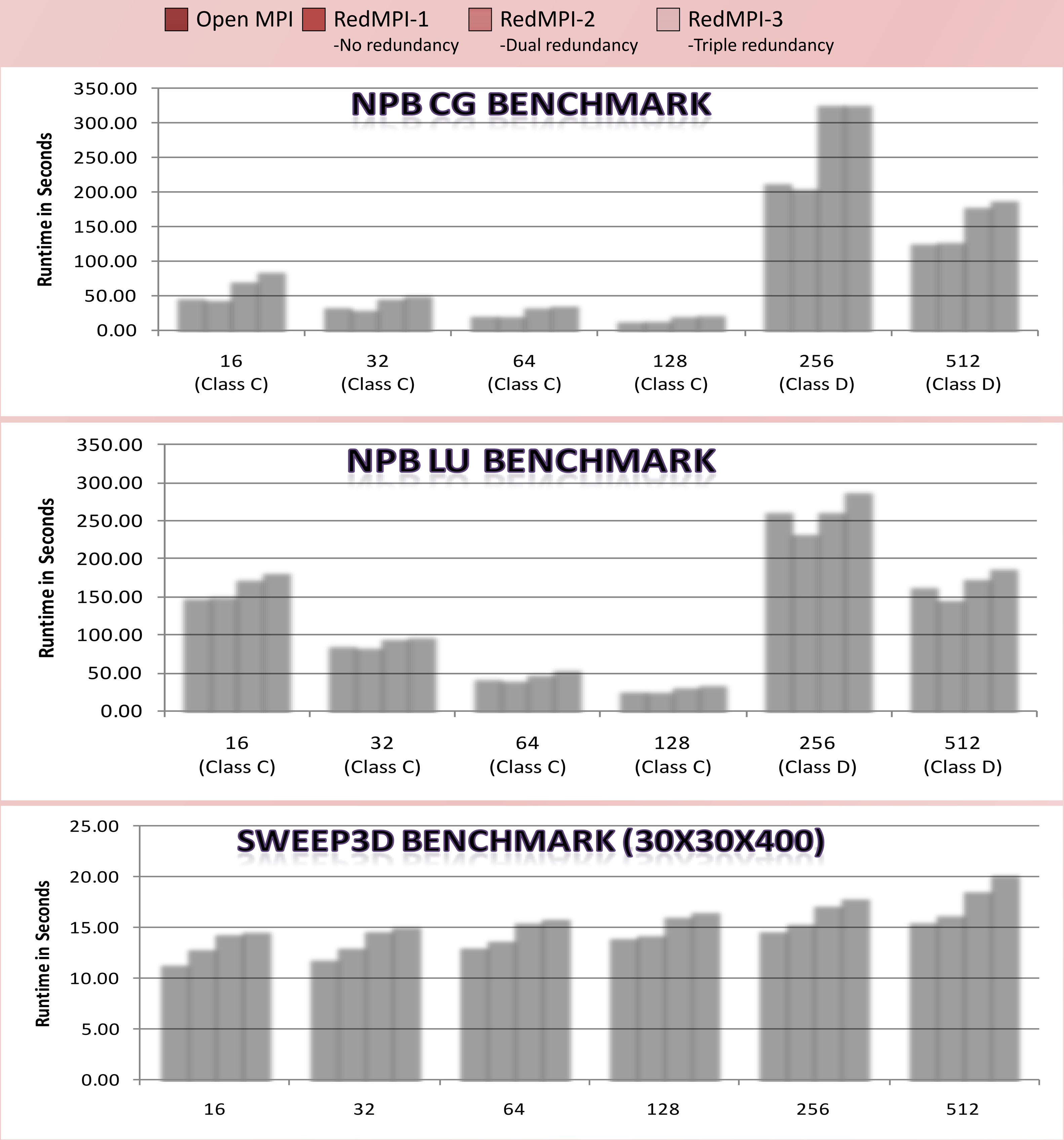


- Enhance performance by sending the original message plus a small hash to a separate replica
  - No longer dependent on *r*r* communication
  - Comparison still performed on receiver-side
  - Hash mismatch triggers secondary voting protocol amongst receiving replicas



Results

- Experiments performed on 96 cluster nodes
  - AMD Opteron 6182 (Magny Core) – 16 cores per node
  - 32GB RAM per node
  - 40Gbit/s Infiniband for MPI Communication
  - Gigabit ethernet for network filesytem



Average Benchmark Overheads with Redundancy

	Dual Redundancy	Triple Redundancy
NPB CG	44%	53%
NPB LU	10%	19%
SWEEP3D	18%	23%

- The cost of triple redundancy is relatively low after dual redundancy
- Redundancy is a viable method to detect and protect from SDCs
- Fault injection experiments successfully demonstrate capacity to detect and correct SDCs in a cluster environment

This work was supported in part by NSF grants CNS-1058779, CNS-0958311, DOE grant DE-FG02-08ER25837, a subcontract from Sandia National Laboratory. and by the Laboratory Directed Research and Development Program of Oak Ridge National Laboratory (ORNL), managed by UT-Battelle, LLC for the U. S. Department of Energy under Contract No. De-AC05-00OR22725.