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

- Naive SDC protection may be achieved by transmitting and comparing $r^r$ messages amongst $r$ total replicas.
  - Induces high interconnect contentention / 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 filesystem

- Runtime in Seconds

- Average Benchmark Overheads with Redundancy

<table>
<thead>
<tr>
<th></th>
<th>Dual Redundancy</th>
<th>Triple Redundancy</th>
</tr>
</thead>
<tbody>
<tr>
<td>NPB CG</td>
<td>44%</td>
<td>53%</td>
</tr>
<tr>
<td>NPB LU</td>
<td>10%</td>
<td>19%</td>
</tr>
<tr>
<td>SWEEP3D</td>
<td>18%</td>
<td>23%</td>
</tr>
</tbody>
</table>

- 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