Bo Fang, Karthik Pattabiraman, Matei Ripeanu

University of British Columbia

1. Motivation

- Large scale systems are prone to experience transient hardware faults
- Main causes include:
  - Cosmic ray
  - Particle strike
  - Voltage fluctuation
- Outcomes:
  - Silent data corruption (undetected incorrect output)
  - Fail-stop failure (terminate unexpectedly)
- Characteristics of recovery techniques:
  - Roll Back
  - Roll Forward

<table>
<thead>
<tr>
<th></th>
<th>Roll Back</th>
<th>Roll Forward</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overhead</td>
<td>$T_f(checkpoint)+T_f(roll)$</td>
<td>$T_f(next state)+T_f(checkpoint)$</td>
</tr>
<tr>
<td>Recovered State</td>
<td>Precise</td>
<td>Precise/Approximate</td>
</tr>
</tbody>
</table>
- Approximate recover may introduce new SDCs!

2. LetGo – A Roll-Forward C/R

- Attempts to avoid reload from checkpoints [1]
- System design
- Evaluation results with fault injections:
  - Without LetGo: 41% SDC, 56% Crash
  - With LetGo: 7%, 22% SDC, 74% Crash
  - LetGo converts 62% crashes into continued execution, with 1% increase in SDC rate and 1% increase in detected rate

3. Goal

- Roll-forward recovery may introduce SDCs
- How likely?
- Current evaluation approach: statistical fault injections
- Problems:
  - Time consuming
  - No indication for a particular repair
  - No predictive power

- Our goal is to efficiently predict how likely a particular repair would lead to a SDC

4. Approach and Challenges

- General idea: tracing the dependent state of the approximate data
- Approach: building dynamic data dependence graph

- Challenge i): State exploration

- Challenge ii): No “ground truth”: approximate recovery implies uncertainty

5. Key Observations

- I. Many HPC applications have repetitive behaviors
  - Control flow divergence detection through profiling
- II. Data convergence
  - Fault masking memory access patterns
- Predicting system: collecting program state and analyzing DDGs

6. Our Design

- The core workflow of the predictor:
- Construct profiled DDGs and affected DDGs
- Match profiled DDGs and affected DDGs
- Examine the DDGs with heuristics
- Check for memory access pattern
- Check for message passing semantics

7. Lesson Learned and Preliminary Results

- Data corruptions across multiple MPI processes
  - Checking if the repair affects message passing is promising
- Size of the affected memory locations:
  - Repair&SDC contain more memory writes than repair&benign
- Match affected DDG and profiled DDG
  - More configurations for profiling size/frequency are needed

---