Scalable Verification of MPI Programs
Anh Vo – Advisor: Ganesh Gopalakrishnan
School of Computing, University of Utah, SLC UT 84112

Problem / Motivation

➢ Large scale MPI applications are hard to verify:
  ➢ Non-determinism makes schedule enforcement hard
  ➢ Bugs appear intermittent, not in all traces
  ➢ Bugs often appear only at large scale
➢ Traditional testing: scalable, but no coverage guarantee over the space of non-determinism
➢ Model checking: guarantees coverage, but not scalable

Need scalable verification tool for non-deterministic MPI programs

Background/Related Work

➢ The current state-of-the-art dynamic verifier, ISP:
  ➢ Verifies MPI programs: deadlocks, resource leaks
  ➢ Guarantees coverage over non-determinism
  ➢ Detects and enforces different execution schedules
  ➢ Example (above):
    ➢ Both P0 & P2 sends can match P1 wildcard receive
    ➢ One match causes error
    ➢ Detect both choices and enforce them:
      ➢ Intercept the wildcard Recv, force receive from P0
      ➢ Restart the program, now force receive from P2

LIMITATIONS:
Verification does not scale beyond dozens of processes, misses bugs that happen only at large scale

Approach/Solution

Algorithm:
➢ Each process keeps a Lamport Clock (currLC)
➢ Increase currLC when observe an event
➢ Each LC is associated with an epoch
➢ Observable events are non-deterministic events
➢ Use piggyback (extra msg) to send LC
➢ Compare incoming m.LC with currLC:
  ➢ If (m.LC < currLC) Then
    ➢ m is "late"; i.e., arriving late
    ➢ check if m can be an alternate match to past events
  ➢ else currLC = max (currLC, m.LC)
➢ Upon program completion, output alternate matches
➢ Schedule Generator generates decisions for rerun (DFS)
➢ Proc's are re-started & follow the decisions in each epoch by replacingRecv(*) w. specificRecv. (GUIDED_RUN)
➢ If new schedules are discovered, explore them as well
➢ Repeat until no more schedules

Running DMA:
➢ Link program with DMA library, execute through mpiexec OR
➢ Within Eclipse through the Graphical Explorer for Message Passing Programs (GEM) plug-in
➢ Provides GUI to invoke ISP and visualizes the results
➢ Can invoke DMA. Visualization of results is in progress

Experimental Results

Scalability-Accuracy Tradeoff

➢ What’s happening:
  ➢ Concurrent wildcard receives in P1 & P2 advance P1 & P2 clocks
  ➢ P2’s msg. technically can be a potential match to R2
  ➢ Concurrent wildcard receives in P1 & P2 advance P1 & P2 clocks
  ➢ P2’s msg. technically can be a potential match to R2
  ➢ LC-piggyback cannot detect this
  ➢ Need Vector-Clock (VC) piggyback
  ➢ does not scale as well as LC
  ➢ Rarely occurs in real benchmarks
  ➢ Most will do some collectives before switching communication pattern
  ➢ LC is sync’ed in some collectives

Acknowledgements

We are thankful to Bronis R. de Supinski, Martin Schulz, and Greg Bronsveldity of Lawrence Livermore National Lab for their support, mentoring and encouragement. We also thank Alan Humphrey, Chris Derrick, and Sriram Aananthakrishnan for their work on GEM.

Supported in part by Microsoft, NSF CNS-0509379, CCF-0811429, CCF-0903408

LIMITATIONS:
Verification does not scale beyond dozens of processes, misses bugs that happen only at large scale