Detecting Concurrency Errors with JPF

http://d3s.mff.cuni.cz

Pavel Parízek

Department of Distributed and Dependable Systems

FACULTY OF MATHEMATICS AND PHYSICS
Charles University
Questions about JPF?
Configurations

- Default JPF: exhaustive search (DFS)
  - Threads scheduled in the order given by their IDs

- Random search order
  \[ +\text{cg.randomize_choices}=\text{VAR\_SEED} \]

- Preemption bounding
  \[ +\text{vm.schedul}e.\text{sync.class}=\text{gov.nasa.jpf.vm.schedule.ContextBoundingSyncPolicy} \]
  \[ +\text{vm.schedul}e.\text{sharedness.class}=\text{gov.nasa.jpf.vm.schedule.ContextBoundingSharednessPolicy} \]
  \[ +\text{contextbound}\_\text{max\_number\_of\_preemptions}=<\text{N}> \]
Configurations

• Breadth first search (BFS)
  
  +search.class=gov.nasa.jpf.search.heuristic.BFSHeuristic
  +search.heuristic.queue_limit=-1

• Maximize thread preemption
  
  +search.class=gov.nasa.jpf.search.heuristic.Interleaving

• Minimize preemption
  
  +search.class=gov.nasa.jpf.search.heuristic.MinimizePreemption

• Maximize blocked threads
  
  +search.class=gov.nasa.jpf.search.heuristic.MostBlocked
What to do now

- Finish remaining tasks from the last seminar
  - Writing reasonable environment for `LinkedList` and `Semaphore` (try different workloads)

- Play with different configurations aiming at efficient detection of concurrency errors
  - Use additional benchmark programs (examples)