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
  + cg.randomize_choices=VAR_SEED

• Preemption bounding
  + vm.schedulersync.class=gov.nasa.jpf.vm.
    schedule.ContextBoundingSyncPolicy
  + vm.schedulersharedness.class=gov.nasa.jpf.vm.
    schedule.ContextBoundingSharednessPolicy
  + contextbound.max_number_of_preemptions=<N>
Configurations

- **Breadth first search (BFS)**
  
  \[+\text{search.class} = \text{gov.nasa.jpf.search.heuristic.BFSHeuristic} + \text{search.heuristic.queue\_limit} = -1\]

- **Maximize thread preemption**
  
  \[+\text{search.class} = \text{gov.nasa.jpf.search.heuristic.Interleaving}\]

- **Minimize preemption**
  
  \[+\text{search.class} = \text{gov.nasa.jpf.search.heuristic.MinimizePreemption}\]

- **Maximize blocked threads**
  
  \[+\text{search.class} = \text{gov.nasa.jpf.search.heuristic.MostBlocked}\]
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)