Detecting Concurrency Errors with JPF

http://d3s.mff.cuni.cz

Pavel Parízek

Department of Distributed and Dependable Systems

CHARLES UNIVERSITY IN PRAGUE

faculty of mathematics and physics
Questions about JPF?
Configurations

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

- Random search order
  - \(+cg\text{-}randomize\text{-}choices=\text{VAR\_SEED}\)

- Preemption bounding
  - \(+\text{vm\_scheduler\_sync\_class=gov\_nasa\_jpf\_vm\_schedule\_ContextBoundingSyncPolicy}\)
  - \(+\text{vm\_scheduler\_sharedness\_class=gov\_nasa\_jpf\_vm\_schedule\_ContextBoundingSharednessPolicy}\)
  - \(+\text{contextbound\_max\_number\_of\_preemptions=<N>}\)
Configurations

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

- **Maximize thread preemption**
  
  ```bash
  +search.class=gov.nasa.jpf.search.heuristic.Interleaving
  ```

- **Minimize preemption**
  
  ```bash
  +search.class=gov.nasa.jpf.search.heuristic.MinimizePreemption
  ```

- **Maximize blocked threads**
  
  ```bash
  +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)
    - http://d3s.mff.cuni.cz/teaching/program_analysis_verification/files/concur_bench.zip