

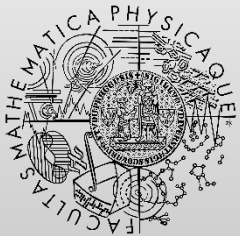
Detecting Concurrency Errors with JPF

<http://d3s.mff.cuni.cz>

Department of
Distributed and
Dependable
Systems



Pavel Parízek



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.scheduler.sync.class=gov.nasa.jpf.vm.  
    schedule.ContextBoundingSyncPolicy  
+vm.scheduler.sharedness.class=gov.nasa.jpf.vm.  
    schedule.ContextBoundingSharednessPolicy  
+contextbound.max_number_of_preemptions=<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)
 - http://d3s.mff.cuni.cz/files/teaching/nswi132/files/concur_bench.zip