

Concurrency meets Bytecode

Aleš Justin

ajustin@redhat.com

Agenda

- Core concurrency concepts
- High level concurrency objects
- Byteman intro and its Arquillian usage

Core concurrency concepts

- Process
- Threads
 - `Runnable::run`
 - `sleep`
 - `yield`
 - `join`

Core concurrency concepts

- Executor
- synchronized keyword
 - instance
 - class
 - object
 - re-entry

Core concurrency concepts

- atomic access
 - all refs, most primitive
 - (excl. long, double)
 - all volatile
- liveness
 - deadlock
 - starvation
 - livelock

Core concurrency concepts

- guarded blocks
- wait (loop!)
- notifyall
- notify
- must have monitor / synchronized!

Core concurrency concepts

- immutable objects
- double checked locking
 - volatile!
- init on demand holder idiom

High Level concurrency objects

- Lock objects
 - Lock, ReadWriteLock, re-entrant
- `java.util.concurrent`
 - Semaphore, CountDownLatch, CyclicBarrier, ...
 - CompletableFuture, Exchanger, Phaser, ...
- Executors
 - service, scheduled, Thread pools
 - Fork/Join

High Level concurrency objects

- Collections
 - BlockingQueue
 - ConcurrentMap
 - ConcurrentNavigableMap
- Check Guava Lib

High Level concurrency objects

- Atomic variables
 - AtomicX
 - Integer, Boolean, Reference, ...
- ThreadLocalRandom
 - Better performance with Fork/Join

Byteman

- Intro

- <http://byteman.jboss.org>

- Arquillian usage

- <https://github.com/GoogleCloudPlatform/appengine-tck/tree/master/core/byteman/src/test/java/com/google/appengine/tck/byteman>

- `JAVA_OPTS="$JAVA_OPTS -Xbootclasspath/p:${BYTEMAN_HOME}/lib/byteman-submit.jar -javaagent:${BYTEMAN_HOME}/lib/byteman.jar=port:9091"`

- <https://docs.oracle.com/javase/tutorial/essential/concurrency/>
- https://en.wikipedia.org/wiki/Double-checked_locking
- <https://github.com/google/guava>

Next CafeBabe

- Finally Jigsaw!
- Tuesday, 15.12.2015 @ 19h, Polygon
- Peter Levart, OpenJDK contributor

Shameless PR

- www.hackathon.si
- Hal mBills
- Saturday / Sunday, 5.-6.12.2015