(base) mimi@mithilas-air test % java -jar target/benchmarks.jar -wi 20 -i 50 -f 2 -tu ns -w 1s -r 1s -bm avgt

# JMH version: 1.37

# VM version: JDK 21.0.4, OpenJDK 64-Bit Server VM, 21.0.4+7-LTS

# VM invoker: /Library/Java/JavaVirtualMachines/temurin-21.jdk/Contents/Home/bin/java

# VM options: <none>

# Blackhole mode: compiler (auto-detected, use -Djmh.blackhole.autoDetect=false to disable)

# Warmup: 20 iterations, 1 s each

# Measurement: 50 iterations, 1 s each

# Timeout: 10 min per iteration

# Threads: 1 thread, will synchronize iterations

# Benchmark mode: Average time, time/op

# Benchmark: org.sample.MyBenchmark.testMethod

# Run progress: 0.00% complete, ETA 00:02:20

# Fork: 1 of 2

# Warmup Iteration 1: 0.560 ns/op

# Warmup Iteration 2: 0.564 ns/op

# Warmup Iteration 3: 0.559 ns/op

# Warmup Iteration 4: 0.559 ns/op

# Warmup Iteration 5: 0.559 ns/op

# Warmup Iteration 6: 0.559 ns/op

# Warmup Iteration 7: 0.559 ns/op

# Warmup Iteration 8: 0.559 ns/op

# Warmup Iteration 9: 0.559 ns/op

# Warmup Iteration 10: 0.559 ns/op

# Warmup Iteration 11: 0.560 ns/op

# Warmup Iteration 12: 0.559 ns/op

# Warmup Iteration 13: 0.559 ns/op

# Warmup Iteration 14: 0.559 ns/op

# Warmup Iteration 15: 0.559 ns/op

# Warmup Iteration 16: 0.561 ns/op

# Warmup Iteration 17: 0.559 ns/op

# Warmup Iteration 18: 0.559 ns/op

# Warmup Iteration 19: 0.559 ns/op

# Warmup Iteration 20: 0.560 ns/op

Iteration 1: 0.562 ns/op

Iteration 2: 0.559 ns/op

Iteration 3: 0.559 ns/op

Iteration 4: 0.724 ns/op

Iteration 5: 0.559 ns/op

Iteration 6: 0.559 ns/op

Iteration 7: 0.559 ns/op

Iteration 8: 0.559 ns/op

Iteration 9: 0.559 ns/op

Iteration 10: 0.561 ns/op

Iteration 11: 0.559 ns/op

Iteration 12: 0.560 ns/op

Iteration 13: 0.560 ns/op

Iteration 14: 0.561 ns/op

Iteration 15: 0.559 ns/op

Iteration 16: 0.559 ns/op

Iteration 17: 0.559 ns/op

Iteration 18: 0.560 ns/op

Iteration 19: 0.563 ns/op

Iteration 20: 0.563 ns/op

Iteration 21: 0.561 ns/op

Iteration 22: 0.561 ns/op

Iteration 23: 0.562 ns/op

Iteration 24: 0.560 ns/op

Iteration 25: 0.559 ns/op

Iteration 26: 0.559 ns/op

Iteration 27: 0.724 ns/op

Iteration 28: 0.563 ns/op

Iteration 29: 0.565 ns/op

Iteration 30: 0.564 ns/op

Iteration 31: 0.567 ns/op

Iteration 32: 0.559 ns/op

Iteration 33: 0.559 ns/op

Iteration 34: 0.559 ns/op

Iteration 35: 0.559 ns/op

Iteration 36: 0.559 ns/op

Iteration 37: 0.559 ns/op

Iteration 38: 0.559 ns/op

Iteration 39: 0.559 ns/op

Iteration 40: 0.559 ns/op

Iteration 41: 0.559 ns/op

Iteration 42: 0.559 ns/op

Iteration 43: 0.657 ns/op

Iteration 44: 0.559 ns/op

Iteration 45: 0.559 ns/op

Iteration 46: 0.559 ns/op

Iteration 47: 0.559 ns/op

Iteration 48: 0.559 ns/op

Iteration 49: 0.559 ns/op

Iteration 50: 0.559 ns/op

# Run progress: 50.00% complete, ETA 00:01:10

# Fork: 2 of 2

# Warmup Iteration 1: 0.560 ns/op

# Warmup Iteration 2: 0.561 ns/op

# Warmup Iteration 3: 0.559 ns/op

# Warmup Iteration 4: 0.559 ns/op

# Warmup Iteration 5: 0.559 ns/op

# Warmup Iteration 6: 0.559 ns/op

# Warmup Iteration 7: 0.559 ns/op

# Warmup Iteration 8: 0.563 ns/op

# Warmup Iteration 9: 0.563 ns/op

# Warmup Iteration 10: 0.562 ns/op

# Warmup Iteration 11: 0.564 ns/op

# Warmup Iteration 12: 0.561 ns/op

# Warmup Iteration 13: 0.559 ns/op

# Warmup Iteration 14: 0.563 ns/op

# Warmup Iteration 15: 0.563 ns/op

# Warmup Iteration 16: 0.563 ns/op

# Warmup Iteration 17: 0.563 ns/op

# Warmup Iteration 18: 0.563 ns/op

# Warmup Iteration 19: 0.713 ns/op

# Warmup Iteration 20: 0.569 ns/op

Iteration 1: 0.583 ns/op

Iteration 2: 0.573 ns/op

Iteration 3: 0.563 ns/op

Iteration 4: 0.564 ns/op

Iteration 5: 0.561 ns/op

Iteration 6: 0.582 ns/op

Iteration 7: 0.606 ns/op

Iteration 8: 0.563 ns/op

Iteration 9: 0.571 ns/op

Iteration 10: 0.591 ns/op

Iteration 11: 0.564 ns/op

Iteration 12: 0.580 ns/op

Iteration 13: 0.603 ns/op

Iteration 14: 0.604 ns/op

Iteration 15: 0.566 ns/op

Iteration 16: 0.561 ns/op

Iteration 17: 0.562 ns/op

Iteration 18: 0.562 ns/op

Iteration 19: 0.564 ns/op

Iteration 20: 0.562 ns/op

Iteration 21: 0.562 ns/op

Iteration 22: 0.561 ns/op

Iteration 23: 0.566 ns/op

Iteration 24: 0.562 ns/op

Iteration 25: 0.563 ns/op

Iteration 26: 0.565 ns/op

Iteration 27: 0.732 ns/op

Iteration 28: 0.565 ns/op

Iteration 29: 0.566 ns/op

Iteration 30: 0.563 ns/op

Iteration 31: 0.563 ns/op

Iteration 32: 0.562 ns/op

Iteration 33: 0.562 ns/op

Iteration 34: 0.563 ns/op

Iteration 35: 0.564 ns/op

Iteration 36: 0.562 ns/op

Iteration 37: 0.563 ns/op

Iteration 38: 0.563 ns/op

Iteration 39: 0.563 ns/op

Iteration 40: 0.564 ns/op

Iteration 41: 0.560 ns/op

Iteration 42: 0.560 ns/op

Iteration 43: 0.564 ns/op

Iteration 44: 0.563 ns/op

Iteration 45: 0.563 ns/op

Iteration 46: 0.563 ns/op

Iteration 47: 0.563 ns/op

Iteration 48: 0.665 ns/op

Iteration 49: 0.562 ns/op

Iteration 50: 0.562 ns/op

Result "org.sample.MyBenchmark.testMethod":

0.571 ±(99.9%) 0.011 ns/op [Average]

(min, avg, max) = (0.559, 0.571, 0.732), stdev = 0.032

CI (99.9%): [0.560, 0.582] (assumes normal distribution)

# Run complete. Total time: 00:02:21

REMEMBER: The numbers below are just data. To gain reusable insights, you need to follow up on

why the numbers are the way they are. Use profilers (see -prof, -lprof), design factorial

experiments, perform baseline and negative tests that provide experimental control, make sure

the benchmarking environment is safe on JVM/OS/HW level, ask for reviews from the domain experts.

Do not assume the numbers tell you what you want them to tell.

NOTE: Current JVM experimentally supports Compiler Blackholes, and they are in use. Please exercise

extra caution when trusting the results, look into the generated code to check the benchmark still

works, and factor in a small probability of new VM bugs. Additionally, while comparisons between

different JVMs are already problematic, the performance difference caused by different Blackhole

modes can be very significant. Please make sure you use the consistent Blackhole mode for comparisons.

Benchmark Mode Cnt Score Error Units

MyBenchmark.testMethod avgt 100 0.571 ± 0.011 ns/op

(base) mimi@mithilas-air test %