```
/Users/zuzannafurtak/Library/Java/JavaVirtualMachines/
corretto-17.0.8/Contents/Home/bin/java -javaagent:/
Applications/IntelliJ IDEA.app/Contents/lib/idea_rt.
jar=59349:/Applications/IntelliJ IDEA.app/Contents/bin
 -Dfile.encoding=UTF-8 -classpath /Users/zuzannafurtak
/studia/las_palmas/big_data/big-data-tasks/assignment-
3/target/classes:/Users/zuzannafurtak/.m2/repository/
org/openjdk/jmh/jmh-core/1.32/jmh-core-1.32.jar:/Users
/zuzannafurtak/.m2/repository/net/sf/jopt-simple/jopt-
simple/4.6/jopt-simple-4.6.jar:/Users/zuzannafurtak/.
m2/repository/org/apache/commons/commons-math3/3.2/
commons-math3-3.2.jar:/Users/zuzannafurtak/.m2/
repository/org/openjdk/jmh/jmh-generator-annprocess/1.
32/jmh-generator-annprocess-1.32.jar com.matrix.
MultiplicationBenchmark
# JMH version: 1.32
# VM version: JDK 17.0.8, OpenJDK 64-Bit Server VM, 17
.0.8+7-LTS
# VM invoker: /Users/zuzannafurtak/Library/Java/
JavaVirtualMachines/corretto-17.0.8/Contents/Home/bin/
java
# VM options: -javaagent:/Applications/IntelliJ IDEA.
app/Contents/lib/idea_rt.jar=59349:/Applications/
IntelliJ IDEA.app/Contents/bin -Dfile.encoding=UTF-8
# Blackhole mode: full + dont-inline hint
# Warmup: 5 iterations, 1 s each
# Measurement: 10 iterations, 1 s each
# Timeout: 10 min per iteration
# Threads: 1 thread, will synchronize iterations
# Benchmark mode: Sampling time
# Benchmark: com.matrix.MultiplicationBenchmark.
multiplyParallelBlocking
# Parameters: (n = 10)

# Run progress: 0.00% complete, ETA 00:02:15
# Fork: 1 of 1
# Warmup Iteration   1: 0.299 ±(99.9%) 0.006 ms/op
# Warmup Iteration   2: 0.275 ±(99.9%) 0.003 ms/op
# Warmup Iteration   3: 0.274 ±(99.9%) 0.003 ms/op
# Warmup Iteration   4: 0.281 ±(99.9%) 0.003 ms/op
# Warmup Iteration   5: 0.278 ±(99.9%) 0.003 ms/op
```

```
Iteration   1: 0.285 ±(99.9%) 0.003 ms/op
                 multiplyParallelBlocking·p0.00:   0.
198 ms/op
                 multiplyParallelBlocking·p0.50:   0.
281 ms/op
                 multiplyParallelBlocking·p0.90:   0.
314 ms/op
                 multiplyParallelBlocking·p0.95:   0.
329 ms/op
                 multiplyParallelBlocking·p0.99:   0.
616 ms/op
                 multiplyParallelBlocking·p0.999:  0.
697 ms/op
                 multiplyParallelBlocking·p0.9999: 0.
731 ms/op
                 multiplyParallelBlocking·p1.00:   0.
731 ms/op

Iteration   2: 0.285 ±(99.9%) 0.003 ms/op
                 multiplyParallelBlocking·p0.00:   0.
191 ms/op
                 multiplyParallelBlocking·p0.50:   0.
279 ms/op
                 multiplyParallelBlocking·p0.90:   0.
317 ms/op
                 multiplyParallelBlocking·p0.95:   0.
334 ms/op
                 multiplyParallelBlocking·p0.99:   0.
626 ms/op
                 multiplyParallelBlocking·p0.999:  0.
745 ms/op
                 multiplyParallelBlocking·p0.9999: 0.
826 ms/op
                 multiplyParallelBlocking·p1.00:   0.
826 ms/op

Iteration   3: 0.285 ±(99.9%) 0.003 ms/op
                 multiplyParallelBlocking·p0.00:   0.
196 ms/op
                 multiplyParallelBlocking·p0.50:   0.
280 ms/op
```

```
                 multiplyParallelBlocking·p0.90:   0.
315 ms/op
                 multiplyParallelBlocking·p0.95:   0.
332 ms/op
                 multiplyParallelBlocking·p0.99:   0.
625 ms/op
                 multiplyParallelBlocking·p0.999:  0.
702 ms/op
                 multiplyParallelBlocking·p0.9999: 0.
774 ms/op
                 multiplyParallelBlocking·p1.00:   0.
774 ms/op

Iteration   4: 0.287 ±(99.9%) 0.003 ms/op
                 multiplyParallelBlocking·p0.00:   0.
190 ms/op
                 multiplyParallelBlocking·p0.50:   0.
282 ms/op
                 multiplyParallelBlocking·p0.90:   0.
317 ms/op
                 multiplyParallelBlocking·p0.95:   0.
335 ms/op
                 multiplyParallelBlocking·p0.99:   0.
630 ms/op
                 multiplyParallelBlocking·p0.999:  0.
741 ms/op
                 multiplyParallelBlocking·p0.9999: 0.
834 ms/op
                 multiplyParallelBlocking·p1.00:   0.
834 ms/op

Iteration   5: 0.288 ±(99.9%) 0.003 ms/op
                 multiplyParallelBlocking·p0.00:   0.
201 ms/op
                 multiplyParallelBlocking·p0.50:   0.
282 ms/op
                 multiplyParallelBlocking·p0.90:   0.
316 ms/op
                 multiplyParallelBlocking·p0.95:   0.
332 ms/op
                 multiplyParallelBlocking·p0.99:   0.
```

```
632 ms/op
                 multiplyParallelBlocking·p0.999:  0.
693 ms/op
                 multiplyParallelBlocking·p0.9999: 0.
703 ms/op
                 multiplyParallelBlocking·p1.00:   0.
703 ms/op

Iteration   6: 0.288 ±(99.9%) 0.003 ms/op
                 multiplyParallelBlocking·p0.00:   0.
191 ms/op
                 multiplyParallelBlocking·p0.50:   0.
283 ms/op
                 multiplyParallelBlocking·p0.90:   0.
318 ms/op
                 multiplyParallelBlocking·p0.95:   0.
334 ms/op
                 multiplyParallelBlocking·p0.99:   0.
633 ms/op
                 multiplyParallelBlocking·p0.999:  0.
747 ms/op
                 multiplyParallelBlocking·p0.9999: 1.
012 ms/op
                 multiplyParallelBlocking·p1.00:   1.
012 ms/op

Iteration   7: 0.289 ±(99.9%) 0.003 ms/op
                 multiplyParallelBlocking·p0.00:   0.
198 ms/op
                 multiplyParallelBlocking·p0.50:   0.
283 ms/op
                 multiplyParallelBlocking·p0.90:   0.
317 ms/op
                 multiplyParallelBlocking·p0.95:   0.
332 ms/op
                 multiplyParallelBlocking·p0.99:   0.
634 ms/op
                 multiplyParallelBlocking·p0.999:  0.
734 ms/op
                 multiplyParallelBlocking·p0.9999: 0.
830 ms/op
```

```
                  multiplyParallelBlocking·p1.00:   0.
830 ms/op

Iteration   8: 0.289 ±(99.9%) 0.003 ms/op
                  multiplyParallelBlocking·p0.00:   0.
196 ms/op
                  multiplyParallelBlocking·p0.50:   0.
283 ms/op
                  multiplyParallelBlocking·p0.90:   0.
318 ms/op
                  multiplyParallelBlocking·p0.95:   0.
337 ms/op
                  multiplyParallelBlocking·p0.99:   0.
639 ms/op
                  multiplyParallelBlocking·p0.999:  0.
732 ms/op
                  multiplyParallelBlocking·p0.9999: 0.
788 ms/op
                  multiplyParallelBlocking·p1.00:   0.
788 ms/op

Iteration   9: 0.288 ±(99.9%) 0.003 ms/op
                  multiplyParallelBlocking·p0.00:   0.
194 ms/op
                  multiplyParallelBlocking·p0.50:   0.
282 ms/op
                  multiplyParallelBlocking·p0.90:   0.
315 ms/op
                  multiplyParallelBlocking·p0.95:   0.
332 ms/op
                  multiplyParallelBlocking·p0.99:   0.
621 ms/op
                  multiplyParallelBlocking·p0.999:  0.
720 ms/op
                  multiplyParallelBlocking·p0.9999: 0.
807 ms/op
                  multiplyParallelBlocking·p1.00:   0.
807 ms/op

Iteration  10: 0.281 ±(99.9%) 0.003 ms/op
                  multiplyParallelBlocking·p0.00:   0.
```

```
191 ms/op
                 multiplyParallelBlocking·p0.50:   0.
275 ms/op
                 multiplyParallelBlocking·p0.90:   0.
313 ms/op
                 multiplyParallelBlocking·p0.95:   0.
330 ms/op
                 multiplyParallelBlocking·p0.99:   0.
627 ms/op
                 multiplyParallelBlocking·p0.999:  0.
741 ms/op
                 multiplyParallelBlocking·p0.9999: 0.
773 ms/op
                 multiplyParallelBlocking·p1.00:   0.
773 ms/op



Result "com.matrix.MultiplicationBenchmark.
multiplyParallelBlocking":
  N = 34999
  mean =      0.286 ±(99.9%) 0.001 ms/op

  Histogram, ms/op:
    [0.100, 0.200) = 31
    [0.200, 0.300) = 27158
    [0.300, 0.400) = 7187
    [0.400, 0.500) = 35
    [0.500, 0.600) = 117
    [0.600, 0.700) = 421
    [0.700, 0.800) = 44
    [0.800, 0.900) = 5
    [0.900, 1.000) = 0

  Percentiles, ms/op:
      p(0.0000) =      0.190 ms/op
     p(50.0000) =      0.281 ms/op
     p(90.0000) =      0.316 ms/op
     p(95.0000) =      0.332 ms/op
     p(99.0000) =      0.626 ms/op
     p(99.9000) =      0.719 ms/op
```

```
     p(99.9900) =      0.832 ms/op
     p(99.9990) =      1.012 ms/op
     p(99.9999) =      1.012 ms/op
    p(100.0000) =      1.012 ms/op


# JMH version: 1.32
# VM version: JDK 17.0.8, OpenJDK 64-Bit Server VM, 17
.0.8+7-LTS
# VM invoker: /Users/zuzannafurtak/Library/Java/
JavaVirtualMachines/corretto-17.0.8/Contents/Home/bin/
java
# VM options: -javaagent:/Applications/IntelliJ IDEA.
app/Contents/lib/idea_rt.jar=59349:/Applications/
IntelliJ IDEA.app/Contents/bin -Dfile.encoding=UTF-8
# Blackhole mode: full + dont-inline hint
# Warmup: 5 iterations, 1 s each
# Measurement: 10 iterations, 1 s each
# Timeout: 10 min per iteration
# Threads: 1 thread, will synchronize iterations
# Benchmark mode: Sampling time
# Benchmark: com.matrix.MultiplicationBenchmark.
multiplyParallelBlocking
# Parameters: (n = 20)

# Run progress: 11.11% complete, ETA 00:02:04
# Fork: 1 of 1
# Warmup Iteration   1: 0.338 ±(99.9%) 0.006 ms/op
# Warmup Iteration   2: 0.324 ±(99.9%) 0.003 ms/op
# Warmup Iteration   3: 0.324 ±(99.9%) 0.003 ms/op
# Warmup Iteration   4: 0.323 ±(99.9%) 0.003 ms/op
# Warmup Iteration   5: 0.321 ±(99.9%) 0.004 ms/op
Iteration   1: 0.321 ±(99.9%) 0.019 ms/op
                 multiplyParallelBlocking·p0.00:   0.
211 ms/op
                 multiplyParallelBlocking·p0.50:   0.
300 ms/op
                 multiplyParallelBlocking·p0.90:   0.
350 ms/op
                 multiplyParallelBlocking·p0.95:   0.
382 ms/op
```

```
                  multiplyParallelBlocking·p0.99:   0.
672 ms/op
                  multiplyParallelBlocking·p0.999:  5.
313 ms/op
                  multiplyParallelBlocking·p0.9999: 14.
287 ms/op
                  multiplyParallelBlocking·p1.00:   14.
287 ms/op

Iteration   2: 0.325 ±(99.9%) 0.005 ms/op
                  multiplyParallelBlocking·p0.00:   0.
217 ms/op
                  multiplyParallelBlocking·p0.50:   0.
317 ms/op
                  multiplyParallelBlocking·p0.90:   0.
355 ms/op
                  multiplyParallelBlocking·p0.95:   0.
375 ms/op
                  multiplyParallelBlocking·p0.99:   0.
673 ms/op
                  multiplyParallelBlocking·p0.999:  0.
906 ms/op
                  multiplyParallelBlocking·p0.9999: 3.
777 ms/op
                  multiplyParallelBlocking·p1.00:   3.
777 ms/op

Iteration   3: 0.308 ±(99.9%) 0.003 ms/op
                  multiplyParallelBlocking·p0.00:   0.
209 ms/op
                  multiplyParallelBlocking·p0.50:   0.
303 ms/op
                  multiplyParallelBlocking·p0.90:   0.
344 ms/op
                  multiplyParallelBlocking·p0.95:   0.
355 ms/op
                  multiplyParallelBlocking·p0.99:   0.
652 ms/op
                  multiplyParallelBlocking·p0.999:  0.
727 ms/op
                  multiplyParallelBlocking·p0.9999: 0.
```

```
748 ms/op
                 multiplyParallelBlocking·p1.00:   0.
748 ms/op

Iteration   4: 0.323 ±(99.9%) 0.003 ms/op
                 multiplyParallelBlocking·p0.00:   0.
231 ms/op
                 multiplyParallelBlocking·p0.50:   0.
316 ms/op
                 multiplyParallelBlocking·p0.90:   0.
351 ms/op
                 multiplyParallelBlocking·p0.95:   0.
366 ms/op
                 multiplyParallelBlocking·p0.99:   0.
666 ms/op
                 multiplyParallelBlocking·p0.999:  0.
765 ms/op
                 multiplyParallelBlocking·p0.9999: 0.
790 ms/op
                 multiplyParallelBlocking·p1.00:   0.
790 ms/op

Iteration   5: 0.317 ±(99.9%) 0.003 ms/op
                 multiplyParallelBlocking·p0.00:   0.
219 ms/op
                 multiplyParallelBlocking·p0.50:   0.
314 ms/op
                 multiplyParallelBlocking·p0.90:   0.
349 ms/op
                 multiplyParallelBlocking·p0.95:   0.
364 ms/op
                 multiplyParallelBlocking·p0.99:   0.
658 ms/op
                 multiplyParallelBlocking·p0.999:  0.
771 ms/op
                 multiplyParallelBlocking·p0.9999: 0.
900 ms/op
                 multiplyParallelBlocking·p1.00:   0.
900 ms/op

Iteration   6: 0.318 ±(99.9%) 0.003 ms/op
```

```
                  multiplyParallelBlocking·p0.00:   0.
226 ms/op
                  multiplyParallelBlocking·p0.50:   0.
314 ms/op
                  multiplyParallelBlocking·p0.90:   0.
354 ms/op
                  multiplyParallelBlocking·p0.95:   0.
369 ms/op
                  multiplyParallelBlocking·p0.99:   0.
658 ms/op
                  multiplyParallelBlocking·p0.999:  0.
786 ms/op
                  multiplyParallelBlocking·p0.9999: 0.
814 ms/op
                  multiplyParallelBlocking·p1.00:   0.
814 ms/op

Iteration   7: 0.430 ±(99.9%) 0.146 ms/op
                  multiplyParallelBlocking·p0.00:   0.
204 ms/op
                  multiplyParallelBlocking·p0.50:   0.
319 ms/op
                  multiplyParallelBlocking·p0.90:   0.
594 ms/op
                  multiplyParallelBlocking·p0.95:   0.
806 ms/op
                  multiplyParallelBlocking·p0.99:   1.
418 ms/op
                  multiplyParallelBlocking·p0.999:  8.
981 ms/op
                  multiplyParallelBlocking·p0.9999: 102
.892 ms/op
                  multiplyParallelBlocking·p1.00:   102
.892 ms/op

Iteration   8: 0.343 ±(99.9%) 0.007 ms/op
                  multiplyParallelBlocking·p0.00:   0.
215 ms/op
                  multiplyParallelBlocking·p0.50:   0.
314 ms/op
                  multiplyParallelBlocking·p0.90:   0.
```

```
448 ms/op
                 multiplyParallelBlocking·p0.95:   0.
532 ms/op
                 multiplyParallelBlocking·p0.99:   0.
797 ms/op
                 multiplyParallelBlocking·p0.999:  1.
355 ms/op
                 multiplyParallelBlocking·p0.9999: 1.
667 ms/op
                 multiplyParallelBlocking·p1.00:   1.
667 ms/op

Iteration   9: 0.312 ±(99.9%) 0.007 ms/op
                 multiplyParallelBlocking·p0.00:   0.
217 ms/op
                 multiplyParallelBlocking·p0.50:   0.
295 ms/op
                 multiplyParallelBlocking·p0.90:   0.
344 ms/op
                 multiplyParallelBlocking·p0.95:   0.
368 ms/op
                 multiplyParallelBlocking·p0.99:   0.
733 ms/op
                 multiplyParallelBlocking·p0.999:  1.
883 ms/op
                 multiplyParallelBlocking·p0.9999: 3.
121 ms/op
                 multiplyParallelBlocking·p1.00:   3.
121 ms/op

Iteration  10: 0.321 ±(99.9%) 0.005 ms/op
                 multiplyParallelBlocking·p0.00:   0.
225 ms/op
                 multiplyParallelBlocking·p0.50:   0.
315 ms/op
                 multiplyParallelBlocking·p0.90:   0.
351 ms/op
                 multiplyParallelBlocking·p0.95:   0.
364 ms/op
                 multiplyParallelBlocking·p0.99:   0.
672 ms/op
```

```
                  multiplyParallelBlocking·p0.999:  1.
709 ms/op
                  multiplyParallelBlocking·p0.9999: 2.
511 ms/op
                  multiplyParallelBlocking·p1.00:   2.
511 ms/op



Result "com.matrix.MultiplicationBenchmark.
multiplyParallelBlocking":
  N = 30564
  mean =      0.329 ±(99.9%) 0.011 ms/op

  Histogram, ms/op:
    [  0.000,  12.500) = 30562
    [ 12.500,  25.000) = 1
    [ 25.000,  37.500) = 0
    [ 37.500,  50.000) = 0
    [ 50.000,  62.500) = 0
    [ 62.500,  75.000) = 0
    [ 75.000,  87.500) = 0
    [ 87.500, 100.000) = 0
    [100.000, 112.500) = 1
    [112.500, 125.000) = 0
    [125.000, 137.500) = 0
    [137.500, 150.000) = 0
    [150.000, 162.500) = 0
    [162.500, 175.000) = 0
    [175.000, 187.500) = 0

  Percentiles, ms/op:
      p(0.0000) =      0.204 ms/op
     p(50.0000) =      0.311 ms/op
     p(90.0000) =      0.358 ms/op
     p(95.0000) =      0.400 ms/op
     p(99.0000) =      0.728 ms/op
     p(99.9000) =      1.727 ms/op
     p(99.9900) =     10.600 ms/op
     p(99.9990) =    102.892 ms/op
     p(99.9999) =    102.892 ms/op
```

```
    p(100.0000) =    102.892 ms/op


# JMH version: 1.32
# VM version: JDK 17.0.8, OpenJDK 64-Bit Server VM, 17
.0.8+7-LTS
# VM invoker: /Users/zuzannafurtak/Library/Java/
JavaVirtualMachines/corretto-17.0.8/Contents/Home/bin/
java
# VM options: -javaagent:/Applications/IntelliJ IDEA.
app/Contents/lib/idea_rt.jar=59349:/Applications/
IntelliJ IDEA.app/Contents/bin -Dfile.encoding=UTF-8
# Blackhole mode: full + dont-inline hint
# Warmup: 5 iterations, 1 s each
# Measurement: 10 iterations, 1 s each
# Timeout: 10 min per iteration
# Threads: 1 thread, will synchronize iterations
# Benchmark mode: Sampling time
# Benchmark: com.matrix.MultiplicationBenchmark.
multiplyParallelBlocking
# Parameters: (n = 50)

# Run progress: 22.22% complete, ETA 00:01:49
# Fork: 1 of 1
# Warmup Iteration   1: 0.346 ±(99.9%) 0.011 ms/op
# Warmup Iteration   2: 0.334 ±(99.9%) 0.003 ms/op
# Warmup Iteration   3: 0.331 ±(99.9%) 0.003 ms/op
# Warmup Iteration   4: 0.327 ±(99.9%) 0.003 ms/op
# Warmup Iteration   5: 0.332 ±(99.9%) 0.003 ms/op
Iteration   1: 0.332 ±(99.9%) 0.004 ms/op
                 multiplyParallelBlocking·p0.00:   0.
249 ms/op
                 multiplyParallelBlocking·p0.50:   0.
324 ms/op
                 multiplyParallelBlocking·p0.90:   0.
360 ms/op
                 multiplyParallelBlocking·p0.95:   0.
374 ms/op
                 multiplyParallelBlocking·p0.99:   0.
669 ms/op
                 multiplyParallelBlocking·p0.999:  0.
```

```
778 ms/op
                 multiplyParallelBlocking·p0.9999: 1.
023 ms/op
                 multiplyParallelBlocking·p1.00:   1.
023 ms/op

Iteration   2: 0.336 ±(99.9%) 0.003 ms/op
                 multiplyParallelBlocking·p0.00:   0.
241 ms/op
                 multiplyParallelBlocking·p0.50:   0.
329 ms/op
                 multiplyParallelBlocking·p0.90:   0.
362 ms/op
                 multiplyParallelBlocking·p0.95:   0.
378 ms/op
                 multiplyParallelBlocking·p0.99:   0.
678 ms/op
                 multiplyParallelBlocking·p0.999:  0.
751 ms/op
                 multiplyParallelBlocking·p0.9999: 0.
776 ms/op
                 multiplyParallelBlocking·p1.00:   0.
776 ms/op

Iteration   3: 0.337 ±(99.9%) 0.003 ms/op
                 multiplyParallelBlocking·p0.00:   0.
248 ms/op
                 multiplyParallelBlocking·p0.50:   0.
328 ms/op
                 multiplyParallelBlocking·p0.90:   0.
365 ms/op
                 multiplyParallelBlocking·p0.95:   0.
382 ms/op
                 multiplyParallelBlocking·p0.99:   0.
681 ms/op
                 multiplyParallelBlocking·p0.999:  0.
761 ms/op
                 multiplyParallelBlocking·p0.9999: 0.
952 ms/op
                 multiplyParallelBlocking·p1.00:   0.
952 ms/op
```

```
Iteration   4: 0.334 ±(99.9%) 0.003 ms/op
                 multiplyParallelBlocking·p0.00:   0.
238 ms/op
                 multiplyParallelBlocking·p0.50:   0.
327 ms/op
                 multiplyParallelBlocking·p0.90:   0.
361 ms/op
                 multiplyParallelBlocking·p0.95:   0.
375 ms/op
                 multiplyParallelBlocking·p0.99:   0.
676 ms/op
                 multiplyParallelBlocking·p0.999:  0.
751 ms/op
                 multiplyParallelBlocking·p0.9999: 0.
783 ms/op
                 multiplyParallelBlocking·p1.00:   0.
783 ms/op

Iteration   5: 0.336 ±(99.9%) 0.003 ms/op
                 multiplyParallelBlocking·p0.00:   0.
258 ms/op
                 multiplyParallelBlocking·p0.50:   0.
328 ms/op
                 multiplyParallelBlocking·p0.90:   0.
362 ms/op
                 multiplyParallelBlocking·p0.95:   0.
377 ms/op
                 multiplyParallelBlocking·p0.99:   0.
668 ms/op
                 multiplyParallelBlocking·p0.999:  0.
773 ms/op
                 multiplyParallelBlocking·p0.9999: 0.
805 ms/op
                 multiplyParallelBlocking·p1.00:   0.
805 ms/op

Iteration   6: 0.337 ±(99.9%) 0.003 ms/op
                 multiplyParallelBlocking·p0.00:   0.
259 ms/op
                 multiplyParallelBlocking·p0.50:   0.
```

```
328 ms/op
                 multiplyParallelBlocking·p0.90:   0.
363 ms/op
                 multiplyParallelBlocking·p0.95:   0.
377 ms/op
                 multiplyParallelBlocking·p0.99:   0.
676 ms/op
                 multiplyParallelBlocking·p0.999:  0.
746 ms/op
                 multiplyParallelBlocking·p0.9999: 0.
809 ms/op
                 multiplyParallelBlocking·p1.00:   0.
809 ms/op

Iteration   7: 0.337 ±(99.9%) 0.003 ms/op
                 multiplyParallelBlocking·p0.00:   0.
262 ms/op
                 multiplyParallelBlocking·p0.50:   0.
329 ms/op
                 multiplyParallelBlocking·p0.90:   0.
364 ms/op
                 multiplyParallelBlocking·p0.95:   0.
380 ms/op
                 multiplyParallelBlocking·p0.99:   0.
671 ms/op
                 multiplyParallelBlocking·p0.999:  0.
736 ms/op
                 multiplyParallelBlocking·p0.9999: 0.
860 ms/op
                 multiplyParallelBlocking·p1.00:   0.
860 ms/op

Iteration   8: 0.335 ±(99.9%) 0.003 ms/op
                 multiplyParallelBlocking·p0.00:   0.
234 ms/op
                 multiplyParallelBlocking·p0.50:   0.
327 ms/op
                 multiplyParallelBlocking·p0.90:   0.
361 ms/op
                 multiplyParallelBlocking·p0.95:   0.
376 ms/op
```

```
                  multiplyParallelBlocking·p0.99:   0.
677 ms/op
                  multiplyParallelBlocking·p0.999:  0.
741 ms/op
                  multiplyParallelBlocking·p0.9999: 0.
876 ms/op
                  multiplyParallelBlocking·p1.00:   0.
876 ms/op

Iteration   9: 0.334 ±(99.9%) 0.003 ms/op
                  multiplyParallelBlocking·p0.00:   0.
228 ms/op
                  multiplyParallelBlocking·p0.50:   0.
327 ms/op
                  multiplyParallelBlocking·p0.90:   0.
362 ms/op
                  multiplyParallelBlocking·p0.95:   0.
376 ms/op
                  multiplyParallelBlocking·p0.99:   0.
669 ms/op
                  multiplyParallelBlocking·p0.999:  0.
736 ms/op
                  multiplyParallelBlocking·p0.9999: 0.
885 ms/op
                  multiplyParallelBlocking·p1.00:   0.
885 ms/op

Iteration  10: 0.337 ±(99.9%) 0.003 ms/op
                  multiplyParallelBlocking·p0.00:   0.
252 ms/op
                  multiplyParallelBlocking·p0.50:   0.
329 ms/op
                  multiplyParallelBlocking·p0.90:   0.
364 ms/op
                  multiplyParallelBlocking·p0.95:   0.
378 ms/op
                  multiplyParallelBlocking·p0.99:   0.
670 ms/op
                  multiplyParallelBlocking·p0.999:  0.
754 ms/op
                  multiplyParallelBlocking·p0.9999: 0.
```

```
810 ms/op
                 multiplyParallelBlocking·p1.00:   0.
810 ms/op



Result "com.matrix.MultiplicationBenchmark.
multiplyParallelBlocking":
  N = 29958
  mean =      0.335 ±(99.9%) 0.001 ms/op

  Histogram, ms/op:
    [0.200, 0.250) = 29
    [0.250, 0.300) = 2989
    [0.300, 0.350) = 21342
    [0.350, 0.400) = 4818
    [0.400, 0.450) = 147
    [0.450, 0.500) = 7
    [0.500, 0.550) = 4
    [0.550, 0.600) = 26
    [0.600, 0.650) = 161
    [0.650, 0.700) = 291
    [0.700, 0.750) = 113
    [0.750, 0.800) = 19
    [0.800, 0.850) = 5
    [0.850, 0.900) = 4
    [0.900, 0.950) = 1
    [0.950, 1.000) = 1
    [1.000, 1.050) = 1

  Percentiles, ms/op:
      p(0.0000) =      0.228 ms/op
     p(50.0000) =      0.328 ms/op
     p(90.0000) =      0.362 ms/op
     p(95.0000) =      0.377 ms/op
     p(99.0000) =      0.671 ms/op
     p(99.9000) =      0.751 ms/op
     p(99.9900) =      0.928 ms/op
     p(99.9990) =      1.023 ms/op
     p(99.9999) =      1.023 ms/op
    p(100.0000) =      1.023 ms/op
```

```
# JMH version: 1.32
# VM version: JDK 17.0.8, OpenJDK 64-Bit Server VM, 17
.0.8+7-LTS
# VM invoker: /Users/zuzannafurtak/Library/Java/
JavaVirtualMachines/corretto-17.0.8/Contents/Home/bin/
java
# VM options: -javaagent:/Applications/IntelliJ IDEA.
app/Contents/lib/idea_rt.jar=59349:/Applications/
IntelliJ IDEA.app/Contents/bin -Dfile.encoding=UTF-8
# Blackhole mode: full + dont-inline hint
# Warmup: 5 iterations, 1 s each
# Measurement: 10 iterations, 1 s each
# Timeout: 10 min per iteration
# Threads: 1 thread, will synchronize iterations
# Benchmark mode: Sampling time
# Benchmark: com.matrix.MultiplicationBenchmark.
multiplyParallelBlocking
# Parameters: (n = 100)

# Run progress: 33.33% complete, ETA 00:01:34
# Fork: 1 of 1
# Warmup Iteration   1: 0.451 ±(99.9%) 0.031 ms/op
# Warmup Iteration   2: 0.410 ±(99.9%) 0.004 ms/op
# Warmup Iteration   3: 0.414 ±(99.9%) 0.004 ms/op
# Warmup Iteration   4: 0.413 ±(99.9%) 0.004 ms/op
# Warmup Iteration   5: 0.412 ±(99.9%) 0.004 ms/op
Iteration   1: 0.413 ±(99.9%) 0.004 ms/op
                 multiplyParallelBlocking·p0.00:   0.
317 ms/op
                 multiplyParallelBlocking·p0.50:   0.
404 ms/op
                 multiplyParallelBlocking·p0.90:   0.
447 ms/op
                 multiplyParallelBlocking·p0.95:   0.
466 ms/op
                 multiplyParallelBlocking·p0.99:   0.
723 ms/op
                 multiplyParallelBlocking·p0.999:  0.
844 ms/op
```

```
                 multiplyParallelBlocking·p0.9999: 0.
922 ms/op
                 multiplyParallelBlocking·p1.00:   0.
922 ms/op

Iteration   2: 0.415 ±(99.9%) 0.004 ms/op
                 multiplyParallelBlocking·p0.00:   0.
313 ms/op
                 multiplyParallelBlocking·p0.50:   0.
407 ms/op
                 multiplyParallelBlocking·p0.90:   0.
450 ms/op
                 multiplyParallelBlocking·p0.95:   0.
467 ms/op
                 multiplyParallelBlocking·p0.99:   0.
728 ms/op
                 multiplyParallelBlocking·p0.999:  0.
850 ms/op
                 multiplyParallelBlocking·p0.9999: 0.
891 ms/op
                 multiplyParallelBlocking·p1.00:   0.
891 ms/op

Iteration   3: 0.483 ±(99.9%) 0.177 ms/op
                 multiplyParallelBlocking·p0.00:   0.
284 ms/op
                 multiplyParallelBlocking·p0.50:   0.
410 ms/op
                 multiplyParallelBlocking·p0.90:   0.
486 ms/op
                 multiplyParallelBlocking·p0.95:   0.
582 ms/op
                 multiplyParallelBlocking·p0.99:   0.
825 ms/op
                 multiplyParallelBlocking·p0.999:  9.
958 ms/op
                 multiplyParallelBlocking·p0.9999: 111
.804 ms/op
                 multiplyParallelBlocking·p1.00:   111
.804 ms/op
```

```
Iteration   4: 0.415 ±(99.9%) 0.005 ms/op
                 multiplyParallelBlocking·p0.00:   0.
318 ms/op
                 multiplyParallelBlocking·p0.50:   0.
401 ms/op
                 multiplyParallelBlocking·p0.90:   0.
462 ms/op
                 multiplyParallelBlocking·p0.95:   0.
492 ms/op
                 multiplyParallelBlocking·p0.99:   0.
807 ms/op
                 multiplyParallelBlocking·p0.999:  0.
928 ms/op
                 multiplyParallelBlocking·p0.9999: 1.
081 ms/op
                 multiplyParallelBlocking·p1.00:   1.
081 ms/op

Iteration   5: 0.413 ±(99.9%) 0.004 ms/op
                 multiplyParallelBlocking·p0.00:   0.
315 ms/op
                 multiplyParallelBlocking·p0.50:   0.
404 ms/op
                 multiplyParallelBlocking·p0.90:   0.
450 ms/op
                 multiplyParallelBlocking·p0.95:   0.
472 ms/op
                 multiplyParallelBlocking·p0.99:   0.
754 ms/op
                 multiplyParallelBlocking·p0.999:  0.
869 ms/op
                 multiplyParallelBlocking·p0.9999: 0.
912 ms/op
                 multiplyParallelBlocking·p1.00:   0.
912 ms/op

Iteration   6: 0.416 ±(99.9%) 0.004 ms/op
                 multiplyParallelBlocking·p0.00:   0.
309 ms/op
                 multiplyParallelBlocking·p0.50:   0.
407 ms/op
```

```
                 multiplyParallelBlocking·p0.90:   0.
451 ms/op
                 multiplyParallelBlocking·p0.95:   0.
470 ms/op
                 multiplyParallelBlocking·p0.99:   0.
749 ms/op
                 multiplyParallelBlocking·p0.999:  0.
809 ms/op
                 multiplyParallelBlocking·p0.9999: 0.
825 ms/op
                 multiplyParallelBlocking·p1.00:   0.
825 ms/op

Iteration   7: 0.388 ±(99.9%) 0.004 ms/op
                 multiplyParallelBlocking·p0.00:   0.
294 ms/op
                 multiplyParallelBlocking·p0.50:   0.
378 ms/op
                 multiplyParallelBlocking·p0.90:   0.
434 ms/op
                 multiplyParallelBlocking·p0.95:   0.
455 ms/op
                 multiplyParallelBlocking·p0.99:   0.
694 ms/op
                 multiplyParallelBlocking·p0.999:  0.
843 ms/op
                 multiplyParallelBlocking·p0.9999: 0.
853 ms/op
                 multiplyParallelBlocking·p1.00:   0.
853 ms/op

Iteration   8: 0.462 ±(99.9%) 0.130 ms/op
                 multiplyParallelBlocking·p0.00:   0.
285 ms/op
                 multiplyParallelBlocking·p0.50:   0.
398 ms/op
                 multiplyParallelBlocking·p0.90:   0.
490 ms/op
                 multiplyParallelBlocking·p0.95:   0.
623 ms/op
                 multiplyParallelBlocking·p0.99:   0.
```

```
811 ms/op
                 multiplyParallelBlocking·p0.999:  9.
756 ms/op
                 multiplyParallelBlocking·p0.9999: 85.
983 ms/op
                 multiplyParallelBlocking·p1.00:   85.
983 ms/op

Iteration   9: 0.407 ±(99.9%) 0.006 ms/op
                 multiplyParallelBlocking·p0.00:   0.
307 ms/op
                 multiplyParallelBlocking·p0.50:   0.
397 ms/op
                 multiplyParallelBlocking·p0.90:   0.
446 ms/op
                 multiplyParallelBlocking·p0.95:   0.
469 ms/op
                 multiplyParallelBlocking·p0.99:   0.
755 ms/op
                 multiplyParallelBlocking·p0.999:  1.
446 ms/op
                 multiplyParallelBlocking·p0.9999: 3.
133 ms/op
                 multiplyParallelBlocking·p1.00:   3.
133 ms/op

Iteration  10: 0.407 ±(99.9%) 0.004 ms/op
                 multiplyParallelBlocking·p0.00:   0.
309 ms/op
                 multiplyParallelBlocking·p0.50:   0.
401 ms/op
                 multiplyParallelBlocking·p0.90:   0.
447 ms/op
                 multiplyParallelBlocking·p0.95:   0.
471 ms/op
                 multiplyParallelBlocking·p0.99:   0.
731 ms/op
                 multiplyParallelBlocking·p0.999:  0.
813 ms/op
                 multiplyParallelBlocking·p0.9999: 0.
835 ms/op
```

```
                    multiplyParallelBlocking·p1.00:   0.
835 ms/op



Result "com.matrix.MultiplicationBenchmark.
multiplyParallelBlocking":
  N = 23933
  mean =      0.420 ±(99.9%) 0.019 ms/op

  Histogram, ms/op:
    [  0.000,  12.500) = 23931
    [ 12.500,  25.000) = 0
    [ 25.000,  37.500) = 0
    [ 37.500,  50.000) = 0
    [ 50.000,  62.500) = 0
    [ 62.500,  75.000) = 0
    [ 75.000,  87.500) = 1
    [ 87.500, 100.000) = 0
    [100.000, 112.500) = 1
    [112.500, 125.000) = 0
    [125.000, 137.500) = 0
    [137.500, 150.000) = 0
    [150.000, 162.500) = 0
    [162.500, 175.000) = 0
    [175.000, 187.500) = 0

  Percentiles, ms/op:
      p(0.0000) =      0.284 ms/op
     p(50.0000) =      0.401 ms/op
     p(90.0000) =      0.453 ms/op
     p(95.0000) =      0.483 ms/op
     p(99.0000) =      0.754 ms/op
     p(99.9000) =      0.936 ms/op
     p(99.9900) =     56.605 ms/op
     p(99.9990) =    111.804 ms/op
     p(99.9999) =    111.804 ms/op
    p(100.0000) =    111.804 ms/op


# JMH version: 1.32
```

```
# VM version: JDK 17.0.8, OpenJDK 64-Bit Server VM, 17
.0.8+7-LTS
# VM invoker: /Users/zuzannafurtak/Library/Java/
JavaVirtualMachines/corretto-17.0.8/Contents/Home/bin/
java
# VM options: -javaagent:/Applications/IntelliJ IDEA.
app/Contents/lib/idea_rt.jar=59349:/Applications/
IntelliJ IDEA.app/Contents/bin -Dfile.encoding=UTF-8
# Blackhole mode: full + dont-inline hint
# Warmup: 5 iterations, 1 s each
# Measurement: 10 iterations, 1 s each
# Timeout: 10 min per iteration
# Threads: 1 thread, will synchronize iterations
# Benchmark mode: Sampling time
# Benchmark: com.matrix.MultiplicationBenchmark.
multiplyParallelBlocking
# Parameters: (n = 200)

# Run progress: 44.44% complete, ETA 00:01:18
# Fork: 1 of 1
# Warmup Iteration   1: 1.192 ±(99.9%) 0.101 ms/op
# Warmup Iteration   2: 1.053 ±(99.9%) 0.011 ms/op
# Warmup Iteration   3: 1.046 ±(99.9%) 0.011 ms/op
# Warmup Iteration   4: 1.068 ±(99.9%) 0.010 ms/op
# Warmup Iteration   5: 1.019 ±(99.9%) 0.008 ms/op
Iteration   1: 1.008 ±(99.9%) 0.008 ms/op
                 multiplyParallelBlocking·p0.00:   0.
901 ms/op
                 multiplyParallelBlocking·p0.50:   0.
990 ms/op
                 multiplyParallelBlocking·p0.90:   1.
077 ms/op
                 multiplyParallelBlocking·p0.95:   1.
118 ms/op
                 multiplyParallelBlocking·p0.99:   1.
352 ms/op
                 multiplyParallelBlocking·p0.999:  1.
485 ms/op
                 multiplyParallelBlocking·p0.9999: 1.
485 ms/op
                 multiplyParallelBlocking·p1.00:   1.
```

```
485 ms/op

Iteration   2: 1.048 ±(99.9%) 0.010 ms/op
                 multiplyParallelBlocking·p0.00:   0.
905 ms/op
                 multiplyParallelBlocking·p0.50:   1.
022 ms/op
                 multiplyParallelBlocking·p0.90:   1.
161 ms/op
                 multiplyParallelBlocking·p0.95:   1.
200 ms/op
                 multiplyParallelBlocking·p0.99:   1.
409 ms/op
                 multiplyParallelBlocking·p0.999:  1.
575 ms/op
                 multiplyParallelBlocking·p0.9999: 1.
575 ms/op
                 multiplyParallelBlocking·p1.00:   1.
575 ms/op

Iteration   3: 1.015 ±(99.9%) 0.008 ms/op
                 multiplyParallelBlocking·p0.00:   0.
905 ms/op
                 multiplyParallelBlocking·p0.50:   0.
995 ms/op
                 multiplyParallelBlocking·p0.90:   1.
087 ms/op
                 multiplyParallelBlocking·p0.95:   1.
159 ms/op
                 multiplyParallelBlocking·p0.99:   1.
377 ms/op
                 multiplyParallelBlocking·p0.999:  1.
532 ms/op
                 multiplyParallelBlocking·p0.9999: 1.
532 ms/op
                 multiplyParallelBlocking·p1.00:   1.
532 ms/op

Iteration   4: 1.004 ±(99.9%) 0.007 ms/op
                 multiplyParallelBlocking·p0.00:   0.
914 ms/op
```

```
                 multiplyParallelBlocking·p0.50:   0.
988 ms/op
                 multiplyParallelBlocking·p0.90:   1.
059 ms/op
                 multiplyParallelBlocking·p0.95:   1.
124 ms/op
                 multiplyParallelBlocking·p0.99:   1.
346 ms/op
                 multiplyParallelBlocking·p0.999:  1.
421 ms/op
                 multiplyParallelBlocking·p0.9999: 1.
421 ms/op
                 multiplyParallelBlocking·p1.00:   1.
421 ms/op

Iteration   5: 1.010 ±(99.9%) 0.008 ms/op
                 multiplyParallelBlocking·p0.00:   0.
899 ms/op
                 multiplyParallelBlocking·p0.50:   0.
991 ms/op
                 multiplyParallelBlocking·p0.90:   1.
098 ms/op
                 multiplyParallelBlocking·p0.95:   1.
163 ms/op
                 multiplyParallelBlocking·p0.99:   1.
341 ms/op
                 multiplyParallelBlocking·p0.999:  1.
552 ms/op
                 multiplyParallelBlocking·p0.9999: 1.
552 ms/op
                 multiplyParallelBlocking·p1.00:   1.
552 ms/op

Iteration   6: 1.268 ±(99.9%) 0.460 ms/op
                 multiplyParallelBlocking·p0.00:   0.
918 ms/op
                 multiplyParallelBlocking·p0.50:   1.
077 ms/op
                 multiplyParallelBlocking·p0.90:   1.
369 ms/op
                 multiplyParallelBlocking·p0.95:   1.
```

```
493 ms/op
                 multiplyParallelBlocking·p0.99:   1.
782 ms/op
                 multiplyParallelBlocking·p0.999:  111
.935 ms/op
                 multiplyParallelBlocking·p0.9999: 111
.935 ms/op
                 multiplyParallelBlocking·p1.00:   111
.935 ms/op

Iteration   7: 1.112 ±(99.9%) 0.014 ms/op
                 multiplyParallelBlocking·p0.00:   0.
905 ms/op
                 multiplyParallelBlocking·p0.50:   1.
098 ms/op
                 multiplyParallelBlocking·p0.90:   1.
257 ms/op
                 multiplyParallelBlocking·p0.95:   1.
362 ms/op
                 multiplyParallelBlocking·p0.99:   1.
519 ms/op
                 multiplyParallelBlocking·p0.999:  1.
845 ms/op
                 multiplyParallelBlocking·p0.9999: 1.
845 ms/op
                 multiplyParallelBlocking·p1.00:   1.
845 ms/op

Iteration   8: 1.047 ±(99.9%) 0.011 ms/op
                 multiplyParallelBlocking·p0.00:   0.
904 ms/op
                 multiplyParallelBlocking·p0.50:   1.
014 ms/op
                 multiplyParallelBlocking·p0.90:   1.
198 ms/op
                 multiplyParallelBlocking·p0.95:   1.
263 ms/op
                 multiplyParallelBlocking·p0.99:   1.
430 ms/op
                 multiplyParallelBlocking·p0.999:  1.
620 ms/op
```

```
                 multiplyParallelBlocking·p0.9999: 1.
620 ms/op
                 multiplyParallelBlocking·p1.00:   1.
620 ms/op

Iteration   9: 0.997 ±(99.9%) 0.007 ms/op
                 multiplyParallelBlocking·p0.00:   0.
898 ms/op
                 multiplyParallelBlocking·p0.50:   0.
985 ms/op
                 multiplyParallelBlocking·p0.90:   1.
041 ms/op
                 multiplyParallelBlocking·p0.95:   1.
068 ms/op
                 multiplyParallelBlocking·p0.99:   1.
327 ms/op
                 multiplyParallelBlocking·p0.999:  1.
425 ms/op
                 multiplyParallelBlocking·p0.9999: 1.
425 ms/op
                 multiplyParallelBlocking·p1.00:   1.
425 ms/op

Iteration  10: 1.070 ±(99.9%) 0.013 ms/op
                 multiplyParallelBlocking·p0.00:   0.
911 ms/op
                 multiplyParallelBlocking·p0.50:   1.
026 ms/op
                 multiplyParallelBlocking·p0.90:   1.
229 ms/op
                 multiplyParallelBlocking·p0.95:   1.
304 ms/op
                 multiplyParallelBlocking·p0.99:   1.
468 ms/op
                 multiplyParallelBlocking·p0.999:  1.
599 ms/op
                 multiplyParallelBlocking·p0.9999: 1.
599 ms/op
                 multiplyParallelBlocking·p1.00:   1.
599 ms/op
```

```
Result "com.matrix.MultiplicationBenchmark.
multiplyParallelBlocking":
  N = 9564
  mean =      1.053 ±(99.9%) 0.038 ms/op

  Histogram, ms/op:
    [  0.000,  12.500) = 9563
    [ 12.500,  25.000) = 0
    [ 25.000,  37.500) = 0
    [ 37.500,  50.000) = 0
    [ 50.000,  62.500) = 0
    [ 62.500,  75.000) = 0
    [ 75.000,  87.500) = 0
    [ 87.500, 100.000) = 0
    [100.000, 112.500) = 1
    [112.500, 125.000) = 0
    [125.000, 137.500) = 0
    [137.500, 150.000) = 0
    [150.000, 162.500) = 0
    [162.500, 175.000) = 0
    [175.000, 187.500) = 0

  Percentiles, ms/op:
      p(0.0000) =      0.898 ms/op
     p(50.0000) =      1.006 ms/op
     p(90.0000) =      1.178 ms/op
     p(95.0000) =      1.262 ms/op
     p(99.0000) =      1.452 ms/op
     p(99.9000) =      1.778 ms/op
     p(99.9900) =    111.935 ms/op
     p(99.9990) =    111.935 ms/op
     p(99.9999) =    111.935 ms/op
    p(100.0000) =    111.935 ms/op


# JMH version: 1.32
# VM version: JDK 17.0.8, OpenJDK 64-Bit Server VM, 17
.0.8+7-LTS
# VM invoker: /Users/zuzannafurtak/Library/Java/
```

```
JavaVirtualMachines/corretto-17.0.8/Contents/Home/bin/
java
# VM options: -javaagent:/Applications/IntelliJ IDEA.
app/Contents/lib/idea_rt.jar=59349:/Applications/
IntelliJ IDEA.app/Contents/bin -Dfile.encoding=UTF-8
# Blackhole mode: full + dont-inline hint
# Warmup: 5 iterations, 1 s each
# Measurement: 10 iterations, 1 s each
# Timeout: 10 min per iteration
# Threads: 1 thread, will synchronize iterations
# Benchmark mode: Sampling time
# Benchmark: com.matrix.MultiplicationBenchmark.
multiplyParallelBlocking
# Parameters: (n = 500)

# Run progress: 55.56% complete, ETA 00:01:02
# Fork: 1 of 1
# Warmup Iteration   1: 16.954 ±(99.9%) 2.335 ms/op
# Warmup Iteration   2: 19.496 ±(99.9%) 0.136 ms/op
# Warmup Iteration   3: 19.283 ±(99.9%) 0.105 ms/op
# Warmup Iteration   4: 19.315 ±(99.9%) 0.097 ms/op
# Warmup Iteration   5: 19.431 ±(99.9%) 0.259 ms/op
Iteration   1: 19.336 ±(99.9%) 0.096 ms/op
                 multiplyParallelBlocking·p0.00:   18.
907 ms/op
                 multiplyParallelBlocking·p0.50:   19.
300 ms/op
                 multiplyParallelBlocking·p0.90:   19.
562 ms/op
                 multiplyParallelBlocking·p0.95:   19.
759 ms/op
                 multiplyParallelBlocking·p0.99:   20.
054 ms/op
                 multiplyParallelBlocking·p0.999:  20.
054 ms/op
                 multiplyParallelBlocking·p0.9999: 20.
054 ms/op
                 multiplyParallelBlocking·p1.00:   20.
054 ms/op

Iteration   2: 19.659 ±(99.9%) 0.336 ms/op
```

```
                  multiplyParallelBlocking·p0.00:   19.
235 ms/op
                  multiplyParallelBlocking·p0.50:   19.
513 ms/op
                  multiplyParallelBlocking·p0.90:   19.
982 ms/op
                  multiplyParallelBlocking·p0.95:   20.
187 ms/op
                  multiplyParallelBlocking·p0.99:   24.
281 ms/op
                  multiplyParallelBlocking·p0.999:  24.
281 ms/op
                  multiplyParallelBlocking·p0.9999: 24.
281 ms/op
                  multiplyParallelBlocking·p1.00:   24.
281 ms/op

Iteration   3: 19.444 ±(99.9%) 0.078 ms/op
                  multiplyParallelBlocking·p0.00:   19.
169 ms/op
                  multiplyParallelBlocking·p0.50:   19.
448 ms/op
                  multiplyParallelBlocking·p0.90:   19.
618 ms/op
                  multiplyParallelBlocking·p0.95:   19.
751 ms/op
                  multiplyParallelBlocking·p0.99:   19.
956 ms/op
                  multiplyParallelBlocking·p0.999:  19.
956 ms/op
                  multiplyParallelBlocking·p0.9999: 19.
956 ms/op
                  multiplyParallelBlocking·p1.00:   19.
956 ms/op

Iteration   4: 19.540 ±(99.9%) 0.144 ms/op
                  multiplyParallelBlocking·p0.00:   19.
169 ms/op
                  multiplyParallelBlocking·p0.50:   19.
497 ms/op
                  multiplyParallelBlocking·p0.90:   19.
```

```
749 ms/op
                 multiplyParallelBlocking·p0.95:   19.
985 ms/op
                 multiplyParallelBlocking·p0.99:   21.
103 ms/op
                 multiplyParallelBlocking·p0.999:  21.
103 ms/op
                 multiplyParallelBlocking·p0.9999: 21.
103 ms/op
                 multiplyParallelBlocking·p1.00:   21.
103 ms/op

Iteration   5: 19.481 ±(99.9%) 0.125 ms/op
                 multiplyParallelBlocking·p0.00:   19.
137 ms/op
                 multiplyParallelBlocking·p0.50:   19.
464 ms/op
                 multiplyParallelBlocking·p0.90:   19.
618 ms/op
                 multiplyParallelBlocking·p0.95:   19.
856 ms/op
                 multiplyParallelBlocking·p0.99:   20.
840 ms/op
                 multiplyParallelBlocking·p0.999:  20.
840 ms/op
                 multiplyParallelBlocking·p0.9999: 20.
840 ms/op
                 multiplyParallelBlocking·p1.00:   20.
840 ms/op

Iteration   6: 19.522 ±(99.9%) 0.260 ms/op
                 multiplyParallelBlocking·p0.00:   19.
169 ms/op
                 multiplyParallelBlocking·p0.50:   19.
399 ms/op
                 multiplyParallelBlocking·p0.90:   19.
798 ms/op
                 multiplyParallelBlocking·p0.95:   20.
595 ms/op
                 multiplyParallelBlocking·p0.99:   22.
577 ms/op
```

```
                  multiplyParallelBlocking·p0.999:  22.
577 ms/op
                  multiplyParallelBlocking·p0.9999: 22.
577 ms/op
                  multiplyParallelBlocking·p1.00:   22.
577 ms/op

Iteration   7: 19.384 ±(99.9%) 0.094 ms/op
                  multiplyParallelBlocking·p0.00:   19.
071 ms/op
                  multiplyParallelBlocking·p0.50:   19.
366 ms/op
                  multiplyParallelBlocking·p0.90:   19.
562 ms/op
                  multiplyParallelBlocking·p0.95:   19.
798 ms/op
                  multiplyParallelBlocking·p0.99:   20.
251 ms/op
                  multiplyParallelBlocking·p0.999:  20.
251 ms/op
                  multiplyParallelBlocking·p0.9999: 20.
251 ms/op
                  multiplyParallelBlocking·p1.00:   20.
251 ms/op

Iteration   8: 19.453 ±(99.9%) 0.073 ms/op
                  multiplyParallelBlocking·p0.00:   19.
202 ms/op
                  multiplyParallelBlocking·p0.50:   19.
415 ms/op
                  multiplyParallelBlocking·p0.90:   19.
739 ms/op
                  multiplyParallelBlocking·p0.95:   19.
771 ms/op
                  multiplyParallelBlocking·p0.99:   19.
825 ms/op
                  multiplyParallelBlocking·p0.999:  19.
825 ms/op
                  multiplyParallelBlocking·p0.9999: 19.
825 ms/op
                  multiplyParallelBlocking·p1.00:   19.
```

```
825 ms/op

Iteration   9: 19.514 ±(99.9%) 0.142 ms/op
                 multiplyParallelBlocking·p0.00:   19.
071 ms/op
                 multiplyParallelBlocking·p0.50:   19.
448 ms/op
                 multiplyParallelBlocking·p0.90:   19.
825 ms/op
                 multiplyParallelBlocking·p0.95:   20.
139 ms/op
                 multiplyParallelBlocking·p0.99:   20.
677 ms/op
                 multiplyParallelBlocking·p0.999:  20.
677 ms/op
                 multiplyParallelBlocking·p0.9999: 20.
677 ms/op
                 multiplyParallelBlocking·p1.00:   20.
677 ms/op

Iteration  10: 19.453 ±(99.9%) 0.094 ms/op
                 multiplyParallelBlocking·p0.00:   19.
071 ms/op
                 multiplyParallelBlocking·p0.50:   19.
464 ms/op
                 multiplyParallelBlocking·p0.90:   19.
684 ms/op
                 multiplyParallelBlocking·p0.95:   19.
818 ms/op
                 multiplyParallelBlocking·p0.99:   20.
218 ms/op
                 multiplyParallelBlocking·p0.999:  20.
218 ms/op
                 multiplyParallelBlocking·p0.9999: 20.
218 ms/op
                 multiplyParallelBlocking·p1.00:   20.
218 ms/op



Result "com.matrix.MultiplicationBenchmark.
```

```
multiplyParallelBlocking":
  N = 522
  mean =     19.478 ±(99.9%) 0.051 ms/op

  Histogram, ms/op:
    [18.000, 18.500) = 0
    [18.500, 19.000) = 1
    [19.000, 19.500) = 363
    [19.500, 20.000) = 140
    [20.000, 20.500) = 12
    [20.500, 21.000) = 2
    [21.000, 21.500) = 2
    [21.500, 22.000) = 0
    [22.000, 22.500) = 0
    [22.500, 23.000) = 1
    [23.000, 23.500) = 0
    [23.500, 24.000) = 0
    [24.000, 24.500) = 1

  Percentiles, ms/op:
      p(0.0000) =     18.907 ms/op
     p(50.0000) =     19.431 ms/op
     p(90.0000) =     19.694 ms/op
     p(95.0000) =     19.852 ms/op
     p(99.0000) =     20.803 ms/op
     p(99.9000) =     24.281 ms/op
     p(99.9900) =     24.281 ms/op
     p(99.9990) =     24.281 ms/op
     p(99.9999) =     24.281 ms/op
    p(100.0000) =     24.281 ms/op


# JMH version: 1.32
# VM version: JDK 17.0.8, OpenJDK 64-Bit Server VM, 17
.0.8+7-LTS
# VM invoker: /Users/zuzannafurtak/Library/Java/
JavaVirtualMachines/corretto-17.0.8/Contents/Home/bin/
java
# VM options: -javaagent:/Applications/IntelliJ IDEA.
app/Contents/lib/idea_rt.jar=59349:/Applications/
IntelliJ IDEA.app/Contents/bin -Dfile.encoding=UTF-8
```

```
# Blackhole mode: full + dont-inline hint
# Warmup: 5 iterations, 1 s each
# Measurement: 10 iterations, 1 s each
# Timeout: 10 min per iteration
# Threads: 1 thread, will synchronize iterations
# Benchmark mode: Sampling time
# Benchmark: com.matrix.MultiplicationBenchmark.
multiplyParallelBlocking
# Parameters: (n = 1000)

# Run progress: 66.67% complete, ETA 00:00:47
# Fork: 1 of 1
# Warmup Iteration   1: 206.569 ±(99.9%) 122.090 ms/op
# Warmup Iteration   2: 188.569 ±(99.9%) 7.578 ms/op
# Warmup Iteration   3: 186.734 ±(99.9%) 6.144 ms/op
# Warmup Iteration   4: 186.079 ±(99.9%) 5.165 ms/op
# Warmup Iteration   5: 186.035 ±(99.9%) 4.641 ms/op
Iteration   1: 188.045 ±(99.9%) 7.030 ms/op
                 multiplyParallelBlocking·p0.00:   184
.287 ms/op
                 multiplyParallelBlocking·p0.50:   188
.088 ms/op
                 multiplyParallelBlocking·p0.90:   191
.627 ms/op
                 multiplyParallelBlocking·p0.95:   191
.627 ms/op
                 multiplyParallelBlocking·p0.99:   191
.627 ms/op
                 multiplyParallelBlocking·p0.999:  191
.627 ms/op
                 multiplyParallelBlocking·p0.9999: 191
.627 ms/op
                 multiplyParallelBlocking·p1.00:   191
.627 ms/op

Iteration   2: 185.816 ±(99.9%) 7.758 ms/op
                 multiplyParallelBlocking·p0.00:   182
.452 ms/op
                 multiplyParallelBlocking·p0.50:   185
.336 ms/op
                 multiplyParallelBlocking·p0.90:   190
```

```
.841 ms/op
                 multiplyParallelBlocking·p0.95:   190
.841 ms/op
                 multiplyParallelBlocking·p0.99:   190
.841 ms/op
                 multiplyParallelBlocking·p0.999:  190
.841 ms/op
                 multiplyParallelBlocking·p0.9999: 190
.841 ms/op
                 multiplyParallelBlocking·p1.00:   190
.841 ms/op

Iteration   3: 188.394 ±(99.9%) 13.015 ms/op
                 multiplyParallelBlocking·p0.00:   184
.287 ms/op
                 multiplyParallelBlocking·p0.50:   186
.384 ms/op
                 multiplyParallelBlocking·p0.90:   196
.608 ms/op
                 multiplyParallelBlocking·p0.95:   196
.608 ms/op
                 multiplyParallelBlocking·p0.99:   196
.608 ms/op
                 multiplyParallelBlocking·p0.999:  196
.608 ms/op
                 multiplyParallelBlocking·p0.9999: 196
.608 ms/op
                 multiplyParallelBlocking·p1.00:   196
.608 ms/op

Iteration   4: 186.909 ±(99.9%) 9.240 ms/op
                 multiplyParallelBlocking·p0.00:   184
.287 ms/op
                 multiplyParallelBlocking·p0.50:   185
.467 ms/op
                 multiplyParallelBlocking·p0.90:   192
.676 ms/op
                 multiplyParallelBlocking·p0.95:   192
.676 ms/op
                 multiplyParallelBlocking·p0.99:   192
.676 ms/op
```

```
                  multiplyParallelBlocking·p0.999:  192
.676 ms/op
                  multiplyParallelBlocking·p0.9999: 192
.676 ms/op
                  multiplyParallelBlocking·p1.00:   192
.676 ms/op

Iteration   5: 187.171 ±(99.9%) 8.735 ms/op
                  multiplyParallelBlocking·p0.00:   183
.763 ms/op
                  multiplyParallelBlocking·p0.50:   186
.647 ms/op
                  multiplyParallelBlocking·p0.90:   191
.365 ms/op
                  multiplyParallelBlocking·p0.95:   191
.365 ms/op
                  multiplyParallelBlocking·p0.99:   191
.365 ms/op
                  multiplyParallelBlocking·p0.999:  191
.365 ms/op
                  multiplyParallelBlocking·p0.9999: 191
.365 ms/op
                  multiplyParallelBlocking·p1.00:   191
.365 ms/op

Iteration   6: 187.608 ±(99.9%) 11.299 ms/op
                  multiplyParallelBlocking·p0.00:   184
.549 ms/op
                  multiplyParallelBlocking·p0.50:   185
.729 ms/op
                  multiplyParallelBlocking·p0.90:   194
.249 ms/op
                  multiplyParallelBlocking·p0.95:   194
.249 ms/op
                  multiplyParallelBlocking·p0.99:   194
.249 ms/op
                  multiplyParallelBlocking·p0.999:  194
.249 ms/op
                  multiplyParallelBlocking·p0.9999: 194
.249 ms/op
                  multiplyParallelBlocking·p1.00:   194
```

```
.249 ms/op

Iteration   7: 185.947 ±(99.9%) 6.144 ms/op
                 multiplyParallelBlocking·p0.00:   184
.287 ms/op
                 multiplyParallelBlocking·p0.50:   185
.205 ms/op
                 multiplyParallelBlocking·p0.90:   190
.054 ms/op
                 multiplyParallelBlocking·p0.95:   190
.054 ms/op
                 multiplyParallelBlocking·p0.99:   190
.054 ms/op
                 multiplyParallelBlocking·p0.999:  190
.054 ms/op
                 multiplyParallelBlocking·p0.9999: 190
.054 ms/op
                 multiplyParallelBlocking·p1.00:   190
.054 ms/op

Iteration   8: 185.205 ±(99.9%) 8.143 ms/op
                 multiplyParallelBlocking·p0.00:   181
.928 ms/op
                 multiplyParallelBlocking·p0.50:   184
.549 ms/op
                 multiplyParallelBlocking·p0.90:   190
.579 ms/op
                 multiplyParallelBlocking·p0.95:   190
.579 ms/op
                 multiplyParallelBlocking·p0.99:   190
.579 ms/op
                 multiplyParallelBlocking·p0.999:  190
.579 ms/op
                 multiplyParallelBlocking·p0.9999: 190
.579 ms/op
                 multiplyParallelBlocking·p1.00:   190
.579 ms/op

Iteration   9: 187.258 ±(99.9%) 14.846 ms/op
                 multiplyParallelBlocking·p0.00:   183
.763 ms/op
```

```
                 multiplyParallelBlocking·p0.50:   185
.336 ms/op
                 multiplyParallelBlocking·p0.90:   197
.919 ms/op
                 multiplyParallelBlocking·p0.95:   197
.919 ms/op
                 multiplyParallelBlocking·p0.99:   197
.919 ms/op
                 multiplyParallelBlocking·p0.999:  197
.919 ms/op
                 multiplyParallelBlocking·p0.9999: 197
.919 ms/op
                 multiplyParallelBlocking·p1.00:   197
.919 ms/op

Iteration  10: 186.734 ±(99.9%) 6.214 ms/op
                 multiplyParallelBlocking·p0.00:   184
.549 ms/op
                 multiplyParallelBlocking·p0.50:   186
.122 ms/op
                 multiplyParallelBlocking·p0.90:   190
.317 ms/op
                 multiplyParallelBlocking·p0.95:   190
.317 ms/op
                 multiplyParallelBlocking·p0.99:   190
.317 ms/op
                 multiplyParallelBlocking·p0.999:  190
.317 ms/op
                 multiplyParallelBlocking·p0.9999: 190
.317 ms/op
                 multiplyParallelBlocking·p1.00:   190
.317 ms/op



Result "com.matrix.MultiplicationBenchmark.
multiplyParallelBlocking":
  N = 60
  mean =    186.909 ±(99.9%) 1.481 ms/op

  Histogram, ms/op:
```

```
    [180.000, 181.250) = 0
    [181.250, 182.500) = 2
    [182.500, 183.750) = 0
    [183.750, 185.000) = 17
    [185.000, 186.250) = 16
    [186.250, 187.500) = 7
    [187.500, 188.750) = 3
    [188.750, 190.000) = 2
    [190.000, 191.250) = 7
    [191.250, 192.500) = 2
    [192.500, 193.750) = 1
    [193.750, 195.000) = 1
    [195.000, 196.250) = 0
    [196.250, 197.500) = 1
    [197.500, 198.750) = 1

  Percentiles, ms/op:
      p(0.0000) =    181.928 ms/op
     p(50.0000) =    185.598 ms/op
     p(90.0000) =    191.339 ms/op
     p(95.0000) =    194.170 ms/op
     p(99.0000) =    197.919 ms/op
     p(99.9000) =    197.919 ms/op
     p(99.9900) =    197.919 ms/op
     p(99.9990) =    197.919 ms/op
     p(99.9999) =    197.919 ms/op
    p(100.0000) =    197.919 ms/op


# JMH version: 1.32
# VM version: JDK 17.0.8, OpenJDK 64-Bit Server VM, 17
.0.8+7-LTS
# VM invoker: /Users/zuzannafurtak/Library/Java/
JavaVirtualMachines/corretto-17.0.8/Contents/Home/bin/
java
# VM options: -javaagent:/Applications/IntelliJ IDEA.
app/Contents/lib/idea_rt.jar=59349:/Applications/
IntelliJ IDEA.app/Contents/bin -Dfile.encoding=UTF-8
# Blackhole mode: full + dont-inline hint
# Warmup: 5 iterations, 1 s each
# Measurement: 10 iterations, 1 s each
```

```
# Timeout: 10 min per iteration
# Threads: 1 thread, will synchronize iterations
# Benchmark mode: Sampling time
# Benchmark: com.matrix.MultiplicationBenchmark.
multiplyParallelBlocking
# Parameters: (n = 1500)

# Run progress: 77.78% complete, ETA 00:00:32
# Fork: 1 of 1
# Warmup Iteration   1: 886.571 ms/op
# Warmup Iteration   2: 856.687 ms/op
# Warmup Iteration   3: 856.687 ms/op
# Warmup Iteration   4: 853.017 ms/op
# Warmup Iteration   5: 849.871 ms/op
Iteration   1: 850.919 ms/op
                 multiplyParallelBlocking·p0.00:   848
.298 ms/op
                 multiplyParallelBlocking·p0.50:   850
.919 ms/op
                 multiplyParallelBlocking·p0.90:   853
.541 ms/op
                 multiplyParallelBlocking·p0.95:   853
.541 ms/op
                 multiplyParallelBlocking·p0.99:   853
.541 ms/op
                 multiplyParallelBlocking·p0.999:  853
.541 ms/op
                 multiplyParallelBlocking·p0.9999: 853
.541 ms/op
                 multiplyParallelBlocking·p1.00:   853
.541 ms/op

Iteration   2: 883.425 ms/op
                 multiplyParallelBlocking·p0.00:   860
.881 ms/op
                 multiplyParallelBlocking·p0.50:   883
.425 ms/op
                 multiplyParallelBlocking·p0.90:   905
.970 ms/op
                 multiplyParallelBlocking·p0.95:   905
.970 ms/op
```

```
                  multiplyParallelBlocking·p0.99:   905
.970 ms/op
                  multiplyParallelBlocking·p0.999:  905
.970 ms/op
                  multiplyParallelBlocking·p0.9999: 905
.970 ms/op
                  multiplyParallelBlocking·p1.00:   905
.970 ms/op

Iteration   3: 855.638 ms/op
                  multiplyParallelBlocking·p0.00:   849
.347 ms/op
                  multiplyParallelBlocking·p0.50:   855
.638 ms/op
                  multiplyParallelBlocking·p0.90:   861
.929 ms/op
                  multiplyParallelBlocking·p0.95:   861
.929 ms/op
                  multiplyParallelBlocking·p0.99:   861
.929 ms/op
                  multiplyParallelBlocking·p0.999:  861
.929 ms/op
                  multiplyParallelBlocking·p0.9999: 861
.929 ms/op
                  multiplyParallelBlocking·p1.00:   861
.929 ms/op

Iteration   4: 851.444 ms/op
                  multiplyParallelBlocking·p0.00:   849
.347 ms/op
                  multiplyParallelBlocking·p0.50:   851
.444 ms/op
                  multiplyParallelBlocking·p0.90:   853
.541 ms/op
                  multiplyParallelBlocking·p0.95:   853
.541 ms/op
                  multiplyParallelBlocking·p0.99:   853
.541 ms/op
                  multiplyParallelBlocking·p0.999:  853
.541 ms/op
                  multiplyParallelBlocking·p0.9999: 853
```

```
.541 ms/op
                 multiplyParallelBlocking·p1.00:   853
.541 ms/op

Iteration   5: 848.822 ms/op
                 multiplyParallelBlocking·p0.00:   848
.298 ms/op
                 multiplyParallelBlocking·p0.50:   848
.822 ms/op
                 multiplyParallelBlocking·p0.90:   849
.347 ms/op
                 multiplyParallelBlocking·p0.95:   849
.347 ms/op
                 multiplyParallelBlocking·p0.99:   849
.347 ms/op
                 multiplyParallelBlocking·p0.999:  849
.347 ms/op
                 multiplyParallelBlocking·p0.9999: 849
.347 ms/op
                 multiplyParallelBlocking·p1.00:   849
.347 ms/op

Iteration   6: 855.114 ms/op
                 multiplyParallelBlocking·p0.00:   853
.541 ms/op
                 multiplyParallelBlocking·p0.50:   855
.114 ms/op
                 multiplyParallelBlocking·p0.90:   856
.687 ms/op
                 multiplyParallelBlocking·p0.95:   856
.687 ms/op
                 multiplyParallelBlocking·p0.99:   856
.687 ms/op
                 multiplyParallelBlocking·p0.999:  856
.687 ms/op
                 multiplyParallelBlocking·p0.9999: 856
.687 ms/op
                 multiplyParallelBlocking·p1.00:   856
.687 ms/op

Iteration   7: 859.832 ms/op
```

```
                  multiplyParallelBlocking·p0.00:   851
.444 ms/op
                  multiplyParallelBlocking·p0.50:   859
.832 ms/op
                  multiplyParallelBlocking·p0.90:   868
.221 ms/op
                  multiplyParallelBlocking·p0.95:   868
.221 ms/op
                  multiplyParallelBlocking·p0.99:   868
.221 ms/op
                  multiplyParallelBlocking·p0.999:  868
.221 ms/op
                  multiplyParallelBlocking·p0.9999: 868
.221 ms/op
                  multiplyParallelBlocking·p1.00:   868
.221 ms/op

Iteration   8: 853.541 ms/op
                  multiplyParallelBlocking·p0.00:   849
.347 ms/op
                  multiplyParallelBlocking·p0.50:   853
.541 ms/op
                  multiplyParallelBlocking·p0.90:   857
.735 ms/op
                  multiplyParallelBlocking·p0.95:   857
.735 ms/op
                  multiplyParallelBlocking·p0.99:   857
.735 ms/op
                  multiplyParallelBlocking·p0.999:  857
.735 ms/op
                  multiplyParallelBlocking·p0.9999: 857
.735 ms/op
                  multiplyParallelBlocking·p1.00:   857
.735 ms/op

Iteration   9: 850.919 ms/op
                  multiplyParallelBlocking·p0.00:   850
.395 ms/op
                  multiplyParallelBlocking·p0.50:   850
.919 ms/op
                  multiplyParallelBlocking·p0.90:   851
```

```
.444 ms/op
                 multiplyParallelBlocking·p0.95:   851
.444 ms/op
                 multiplyParallelBlocking·p0.99:   851
.444 ms/op
                 multiplyParallelBlocking·p0.999:  851
.444 ms/op
                 multiplyParallelBlocking·p0.9999: 851
.444 ms/op
                 multiplyParallelBlocking·p1.00:   851
.444 ms/op

Iteration  10: 853.017 ms/op
                 multiplyParallelBlocking·p0.00:   849
.347 ms/op
                 multiplyParallelBlocking·p0.50:   853
.017 ms/op
                 multiplyParallelBlocking·p0.90:   856
.687 ms/op
                 multiplyParallelBlocking·p0.95:   856
.687 ms/op
                 multiplyParallelBlocking·p0.99:   856
.687 ms/op
                 multiplyParallelBlocking·p0.999:  856
.687 ms/op
                 multiplyParallelBlocking·p0.9999: 856
.687 ms/op
                 multiplyParallelBlocking·p1.00:   856
.687 ms/op



Result "com.matrix.MultiplicationBenchmark.
multiplyParallelBlocking":
  N = 20
  mean =    856.267 ±(99.9%) 11.161 ms/op

  Histogram, ms/op:
    [840.000, 845.000) = 0
    [845.000, 850.000) = 7
    [850.000, 855.000) = 6
```

```
    [855.000, 860.000) = 3
    [860.000, 865.000) = 2
    [865.000, 870.000) = 1
    [870.000, 875.000) = 0
    [875.000, 880.000) = 0
    [880.000, 885.000) = 0
    [885.000, 890.000) = 0
    [890.000, 895.000) = 0
    [895.000, 900.000) = 0
    [900.000, 905.000) = 0

  Percentiles, ms/op:
      p(0.0000) =    848.298 ms/op
     p(50.0000) =    852.492 ms/op
     p(90.0000) =    867.592 ms/op
     p(95.0000) =    904.082 ms/op
     p(99.0000) =    905.970 ms/op
     p(99.9000) =    905.970 ms/op
     p(99.9900) =    905.970 ms/op
     p(99.9990) =    905.970 ms/op
     p(99.9999) =    905.970 ms/op
    p(100.0000) =    905.970 ms/op


# JMH version: 1.32
# VM version: JDK 17.0.8, OpenJDK 64-Bit Server VM, 17
.0.8+7-LTS
# VM invoker: /Users/zuzannafurtak/Library/Java/
JavaVirtualMachines/corretto-17.0.8/Contents/Home/bin/
java
# VM options: -javaagent:/Applications/IntelliJ IDEA.
app/Contents/lib/idea_rt.jar=59349:/Applications/
IntelliJ IDEA.app/Contents/bin -Dfile.encoding=UTF-8
# Blackhole mode: full + dont-inline hint
# Warmup: 5 iterations, 1 s each
# Measurement: 10 iterations, 1 s each
# Timeout: 10 min per iteration
# Threads: 1 thread, will synchronize iterations
# Benchmark mode: Sampling time
# Benchmark: com.matrix.MultiplicationBenchmark.
multiplyParallelBlocking
```

```
# Parameters: (n = 2000)

# Run progress: 88.89% complete, ETA 00:00:17
# Fork: 1 of 1
# Warmup Iteration   1: 2302.673 ms/op
# Warmup Iteration   2: 2273.313 ms/op
# Warmup Iteration   3: 2235.564 ms/op
# Warmup Iteration   4: 2248.147 ms/op
# Warmup Iteration   5: 2235.564 ms/op
Iteration   1: 2239.758 ms/op
                 multiplyParallelBlocking·p0.00:
2239.758 ms/op
                 multiplyParallelBlocking·p0.50:
2239.758 ms/op
                 multiplyParallelBlocking·p0.90:
2239.758 ms/op
                 multiplyParallelBlocking·p0.95:
2239.758 ms/op
                 multiplyParallelBlocking·p0.99:
2239.758 ms/op
                 multiplyParallelBlocking·p0.999:
2239.758 ms/op
                 multiplyParallelBlocking·p0.9999:
2239.758 ms/op
                 multiplyParallelBlocking·p1.00:
2239.758 ms/op

Iteration   2: 2243.953 ms/op
                 multiplyParallelBlocking·p0.00:
2243.953 ms/op
                 multiplyParallelBlocking·p0.50:
2243.953 ms/op
                 multiplyParallelBlocking·p0.90:
2243.953 ms/op
                 multiplyParallelBlocking·p0.95:
2243.953 ms/op
                 multiplyParallelBlocking·p0.99:
2243.953 ms/op
                 multiplyParallelBlocking·p0.999:
2243.953 ms/op
                 multiplyParallelBlocking·p0.9999:
```

```
2243.953 ms/op
                 multiplyParallelBlocking·p1.00:
2243.953 ms/op

Iteration   3: 2231.370 ms/op
                 multiplyParallelBlocking·p0.00:
2231.370 ms/op
                 multiplyParallelBlocking·p0.50:
2231.370 ms/op
                 multiplyParallelBlocking·p0.90:
2231.370 ms/op
                 multiplyParallelBlocking·p0.95:
2231.370 ms/op
                 multiplyParallelBlocking·p0.99:
2231.370 ms/op
                 multiplyParallelBlocking·p0.999:
2231.370 ms/op
                 multiplyParallelBlocking·p0.9999:
2231.370 ms/op
                 multiplyParallelBlocking·p1.00:
2231.370 ms/op

Iteration   4: 2222.981 ms/op
                 multiplyParallelBlocking·p0.00:
2222.981 ms/op
                 multiplyParallelBlocking·p0.50:
2222.981 ms/op
                 multiplyParallelBlocking·p0.90:
2222.981 ms/op
                 multiplyParallelBlocking·p0.95:
2222.981 ms/op
                 multiplyParallelBlocking·p0.99:
2222.981 ms/op
                 multiplyParallelBlocking·p0.999:
2222.981 ms/op
                 multiplyParallelBlocking·p0.9999:
2222.981 ms/op
                 multiplyParallelBlocking·p1.00:
2222.981 ms/op

Iteration   5: 2214.593 ms/op
```

```
                  multiplyParallelBlocking·p0.00:
2214.593 ms/op
                  multiplyParallelBlocking·p0.50:
2214.593 ms/op
                  multiplyParallelBlocking·p0.90:
2214.593 ms/op
                  multiplyParallelBlocking·p0.95:
2214.593 ms/op
                  multiplyParallelBlocking·p0.99:
2214.593 ms/op
                  multiplyParallelBlocking·p0.999:
2214.593 ms/op
                  multiplyParallelBlocking·p0.9999:
2214.593 ms/op
                  multiplyParallelBlocking·p1.00:
2214.593 ms/op

Iteration   6: 2193.621 ms/op
                  multiplyParallelBlocking·p0.00:
2193.621 ms/op
                  multiplyParallelBlocking·p0.50:
2193.621 ms/op
                  multiplyParallelBlocking·p0.90:
2193.621 ms/op
                  multiplyParallelBlocking·p0.95:
2193.621 ms/op
                  multiplyParallelBlocking·p0.99:
2193.621 ms/op
                  multiplyParallelBlocking·p0.999:
2193.621 ms/op
                  multiplyParallelBlocking·p0.9999:
2193.621 ms/op
                  multiplyParallelBlocking·p1.00:
2193.621 ms/op

Iteration   7: 2210.398 ms/op
                  multiplyParallelBlocking·p0.00:
2210.398 ms/op
                  multiplyParallelBlocking·p0.50:
2210.398 ms/op
                  multiplyParallelBlocking·p0.90:
```

```
2210.398 ms/op
                 multiplyParallelBlocking·p0.95:
2210.398 ms/op
                 multiplyParallelBlocking·p0.99:
2210.398 ms/op
                 multiplyParallelBlocking·p0.999:
2210.398 ms/op
                 multiplyParallelBlocking·p0.9999:
2210.398 ms/op
                 multiplyParallelBlocking·p1.00:
2210.398 ms/op

Iteration   8: 2202.010 ms/op
                 multiplyParallelBlocking·p0.00:
2202.010 ms/op
                 multiplyParallelBlocking·p0.50:
2202.010 ms/op
                 multiplyParallelBlocking·p0.90:
2202.010 ms/op
                 multiplyParallelBlocking·p0.95:
2202.010 ms/op
                 multiplyParallelBlocking·p0.99:
2202.010 ms/op
                 multiplyParallelBlocking·p0.999:
2202.010 ms/op
                 multiplyParallelBlocking·p0.9999:
2202.010 ms/op
                 multiplyParallelBlocking·p1.00:
2202.010 ms/op

Iteration   9: 2399.142 ms/op
                 multiplyParallelBlocking·p0.00:
2399.142 ms/op
                 multiplyParallelBlocking·p0.50:
2399.142 ms/op
                 multiplyParallelBlocking·p0.90:
2399.142 ms/op
                 multiplyParallelBlocking·p0.95:
2399.142 ms/op
                 multiplyParallelBlocking·p0.99:
2399.142 ms/op
```

```
                  multiplyParallelBlocking·p0.999:
2399.142 ms/op
                  multiplyParallelBlocking·p0.9999:
2399.142 ms/op
                  multiplyParallelBlocking·p1.00:
2399.142 ms/op

Iteration  10: 2424.308 ms/op
                  multiplyParallelBlocking·p0.00:
2424.308 ms/op
                  multiplyParallelBlocking·p0.50:
2424.308 ms/op
                  multiplyParallelBlocking·p0.90:
2424.308 ms/op
                  multiplyParallelBlocking·p0.95:
2424.308 ms/op
                  multiplyParallelBlocking·p0.99:
2424.308 ms/op
                  multiplyParallelBlocking·p0.999:
2424.308 ms/op
                  multiplyParallelBlocking·p0.9999:
2424.308 ms/op
                  multiplyParallelBlocking·p1.00:
2424.308 ms/op



Result "com.matrix.MultiplicationBenchmark.
multiplyParallelBlocking":
  N = 10
  mean =   2258.213 ±(99.9%) 124.950 ms/op

  Histogram, ms/op:
    [2100.000, 2125.000) = 0
    [2125.000, 2150.000) = 0
    [2150.000, 2175.000) = 0
    [2175.000, 2200.000) = 1
    [2200.000, 2225.000) = 4
    [2225.000, 2250.000) = 3
    [2250.000, 2275.000) = 0
    [2275.000, 2300.000) = 0
```

```
     [2300.000, 2325.000) = 0
     [2325.000, 2350.000) = 0
     [2350.000, 2375.000) = 0
     [2375.000, 2400.000) = 1
     [2400.000, 2425.000) = 1
     [2425.000, 2450.000) = 0
     [2450.000, 2475.000) = 0

  Percentiles, ms/op:
      p(0.0000) =   2193.621 ms/op
     p(50.0000) =   2227.175 ms/op
     p(90.0000) =   2421.791 ms/op
     p(95.0000) =   2424.308 ms/op
     p(99.0000) =   2424.308 ms/op
     p(99.9000) =   2424.308 ms/op
     p(99.9900) =   2424.308 ms/op
     p(99.9990) =   2424.308 ms/op
     p(99.9999) =   2424.308 ms/op
    p(100.0000) =   2424.308 ms/op


# Run complete. Total time: 00:02:54

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.

Benchmark

                      (n)    Mode    Cnt     Score
Error  Units
MultiplicationBenchmark.multiplyParallelBlocking
                                       10  sample  34999
     0.286 ±   0.001  ms/op
```

```
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p0.00      10  sample
            0.190            ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p0.50      10  sample
            0.281            ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p0.90      10  sample
            0.316            ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p0.95      10  sample
            0.332            ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p0.99      10  sample
            0.626            ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p0.999     10  sample
            0.719            ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p0.9999    10  sample
            0.832            ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p1.00      10  sample
            1.012            ms/op
MultiplicationBenchmark.multiplyParallelBlocking
                                     20  sample  30564
     0.329 ±   0.011  ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p0.00      20  sample
            0.204            ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p0.50      20  sample
            0.311            ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p0.90      20  sample
            0.358            ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p0.95      20  sample
            0.400            ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p0.99      20  sample
```

```
             0.728            ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p0.999     20  sample
             1.727            ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p0.9999    20  sample
            10.600            ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p1.00      20  sample
           102.892            ms/op
MultiplicationBenchmark.multiplyParallelBlocking
                                     50  sample  29958
     0.335 ±   0.001  ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p0.00      50  sample
             0.228            ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p0.50      50  sample
             0.328            ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p0.90      50  sample
             0.362            ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p0.95      50  sample
             0.377            ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p0.99      50  sample
             0.671            ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p0.999     50  sample
             0.751            ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p0.9999    50  sample
             0.928            ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p1.00      50  sample
             1.023            ms/op
MultiplicationBenchmark.multiplyParallelBlocking
                                    100  sample  23933
     0.420 ±   0.019  ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
```

```
multiplyParallelBlocking·p0.00     100  sample
            0.284            ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p0.50     100  sample
            0.401            ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p0.90     100  sample
            0.453            ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p0.95     100  sample
            0.483            ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p0.99     100  sample
            0.754            ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p0.999    100  sample
            0.936            ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p0.9999   100  sample
           56.605            ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p1.00     100  sample
          111.804            ms/op
MultiplicationBenchmark.multiplyParallelBlocking
                                     200  sample   9564
     1.053 ±   0.038  ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p0.00     200  sample
            0.898            ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p0.50     200  sample
            1.006            ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p0.90     200  sample
            1.178            ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p0.95     200  sample
            1.262            ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p0.99     200  sample
            1.452            ms/op
```

```
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p0.999    200  sample
            1.778            ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p0.9999   200  sample
          111.935            ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p1.00     200  sample
          111.935            ms/op
MultiplicationBenchmark.multiplyParallelBlocking
                                     500  sample    522
    19.478 ±   0.051  ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p0.00     500  sample
           18.907            ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p0.50     500  sample
           19.431            ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p0.90     500  sample
           19.694            ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p0.95     500  sample
           19.852            ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p0.99     500  sample
           20.803            ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p0.999    500  sample
           24.281            ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p0.9999   500  sample
           24.281            ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p1.00     500  sample
           24.281            ms/op
MultiplicationBenchmark.multiplyParallelBlocking
                                    1000  sample     60
   186.909 ±   1.481  ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p0.00    1000  sample
```

```
          181.928            ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p0.50    1000  sample
          185.598            ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p0.90    1000  sample
          191.339            ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p0.95    1000  sample
          194.170            ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p0.99    1000  sample
          197.919            ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p0.999   1000  sample
          197.919            ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p0.9999  1000  sample
          197.919            ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p1.00    1000  sample
          197.919            ms/op
MultiplicationBenchmark.multiplyParallelBlocking
                                     1500  sample     20
   856.267 ±  11.161  ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p0.00    1500  sample
          848.298            ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p0.50    1500  sample
          852.492            ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p0.90    1500  sample
          867.592            ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p0.95    1500  sample
          904.082            ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p0.99    1500  sample
          905.970            ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
```

```
multiplyParallelBlocking·p0.999   1500  sample
          905.970            ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p0.9999  1500  sample
          905.970            ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p1.00    1500  sample
          905.970            ms/op
MultiplicationBenchmark.multiplyParallelBlocking
                                     2000  sample     10
  2258.213 ± 124.950  ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p0.00    2000  sample
         2193.621            ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p0.50    2000  sample
         2227.175            ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p0.90    2000  sample
         2421.791            ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p0.95    2000  sample
         2424.308            ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p0.99    2000  sample
         2424.308            ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p0.999   2000  sample
         2424.308            ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p0.9999  2000  sample
         2424.308            ms/op
MultiplicationBenchmark.multiplyParallelBlocking:
multiplyParallelBlocking·p1.00    2000  sample
         2424.308            ms/op

Process finished with exit code 0
```