Grunnatriði stýrikerfa - dæmatímaverkefni 3

January 2017

Processors: 4
Solutions:
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
All done
Total time: 1028 ms

1. Sequentially: Don't run the next instance until the one before has returned

Additions:

```
for (int i = 0; i < NUMBER_OF_PROBLEMS; i++) {
    Solver.findAndPrintSolution(Problematic.nextProblem());
}
```

Output:

```
Processors: 4
Solutions:
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (461,73) - (99,155) - length: 444 - iterations: 204752
Problem: (615,254) - (394,470) - length: 437 - iterations: 188783
Problem: (296,109) - (34,279) - length: 432 - iterations: 243092
Problem: (85,493) - (61,440) - length: 77 - iterations: 11811
Problem: (263,29) - (336,608) - length: 652 - iterations: 329093
Problem: (279,152) - (618,465) - length: 652 - iterations: 383761
Problem: (276,214) - (2,334) - length: 394 - iterations: 263061
Problem: (113,34) - (489,423) - length: 765 - iterations: 342223
Problem: (60,307) - (350,354) - length: 337 - iterations: 149614
Problem: (393,490) - (64,8) - length: 811 - iterations: 406964
Problem: (493,167) - (222,265) - length: 369 - iterations: 183582
Problem: (422,584) - (107,589) - length: 320 - iterations: 125153
Problem: (216,64) - (525,344) - length: 589 - iterations: 304957
Problem: (626,79) - (148,152) - length: 551 - iterations: 200691
Problem: (303,611) - (328,206) - length: 430 - iterations: 191063
Problem: (313,267) - (603,279) - length: 302 - iterations: 181243
Problem: (232,148) - (381,22) - length: 275 - iterations: 133651
Problem: (423,194) - (398,320) - length: 151 - iterations: 45554
Problem: (119,429) - (111,580) - length: 159 - iterations: 48948
Problem: (13,309) - (113,149) - length: 260 - iterations: 74511
Problem: (279,35) - (623,21) - length: 358 - iterations: 147055
Problem: (413,605) - (248,223) - length: 547 - iterations: 260947
Problem: (130,454) - (548,561) - length: 525 - iterations: 296899
```

Problem: (297,432) - (392,501) - length: 164 - iterations: 53931
Problem: (436,382) - (324,618) - length: 348 - iterations: 213150
Problem: (359,475) - (441,372) - length: 185 - iterations: 68236
Problem: (299,273) - (344,252) - length: 66 - iterations: 8754
Problem: (431,263) - (150,389) - length: 407 - iterations: 270185
Problem: (553,90) - (379,333) - length: 417 - iterations: 160783
All done
Total time: 24009 ms

Comments:

The output was in the same order, except it ran the problem 30 times, The total time was about 24 times as long which leads me to assume that there is a certain amount of constant time involved maybe to do with printing the output, so that the time isn't completely linear.

2. All at once in separate threads. A new thread is created for each instance.

Additions:

In Main Class:

```
final Problem problem = getProblem();

for (int i = 0; i < NUMBER_OF_PROBLEMS; i++) {
        runProblem(makeRunnable(problem));
}
```

Helper Classes:

```
  static Problem getProblem() {
        return Problematic.nextProblem();
  }

  static void runProblem(Runnable runnable) {
        new Thread(runnable).start();
  }

  static Runnable makeRunnable(Problem problem) {

        Runnable runnable = (new Runnable() {
              @Override
              public void run() {
                    Solver.findAndPrintSolution(problem);
              }});

              return runnable;
  }
```

Output:

Processors: 4

```
Solutions:
All done
Total time: 219 ms
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
```

Comments:

This time the total time was outputted before the problems started to printout, which implies that the for loop was about to start a thread and move on the the next thread before one of the problems was able to execute. So there was a moment where 30 threads were running simultaneously, before the first was able to finish. And because Problem.nextProblem() was constant, and only run once that cut the runtime it would have taken to run 30 times.

3. A certain number at a time. Threads are run through a thread pool of a certain size (students can try different sizes). New instances aren’t run until a thread is free in the thread pool.

Addition:

```
ExecutorService threadPool = Executors.newFixedThreadPool(POOL_SIZE);
```

```
final Problem problem = getProblem();

for (int i = 0; i < NUMBER_OF_PROBLEMS; i++) {
    Runnable runnable = makeRunnable(problem);
    threadPool.execute(runnable);
}

try {
    threadPool.shutdown();
    threadPool.awaitTermination(5, TimeUnit.MINUTES);
} catch (InterruptedException e) {
    e.printStackTrace();
}
```

Output:

Pool size 2:

```
Processors: 4
Solutions:
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
```

All done
Total time: 13981 ms

Pool size 10:

Processors: 4
Solutions:
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
Problem: (557,160) - (110,177) - length: 464 - iterations: 217040
All done
Total time: 7037 ms

Comments:

Using the thread pool the total time has increased because all the threads aren't being run at the same time, but when the pool sizes increases the total time is reduced,

because more threads are allowed to work at the same time. As you can see from the output increasing the amount of threads from 2 to 10 more than halved the runtime. I observed the same effects when I increased the number of instances as well.