Chandrakanth Chittappa (001300076)

INFO 6205

Program Structures & Algorithms

Fall 2021

Assignment 5

Please see the presentation on Assignment on Parallel Sorting under the Exams. etc. module.

Your task is to implement a parallel sorting algorithm such that each partition of the array is sorted in parallel. You will consider two different schemes for deciding whether to sort in parallel.

1. A cutoff (defaults to, say, 1000) which you will update according to the first argument in the command line when running. It's your job to experiment and come up with a good value for this cutoff. If there are fewer elements to sort than the cutoff, then you should use the system sort instead.
2. Recursion depth or the number of available threads. Using this determination, you might decide on an ideal number (t) of separate threads (stick to powers of 2) and arrange for that number of partitions to be parallelized (by preventing recursion after the depth of lg t is reached).
3. An appropriate combination of these.

There is a Main class and the ParSort class in the sort.par package of the INFO6205 repository. The Main class can be used as is but the ParSort class needs to be implemented where you see "TODO..." [it turns out that these TODOs are already implemented].

Unless you have a good reason not to, you should just go along with the Java8-style future implementations provided for you in the class repository.

You must prepare a report that shows the results of your experiments and draws a conclusion (or more) about the efficacy of this method of parallelizing sort. Your experiments should involve sorting arrays of sufficient size for the parallel sort to make a difference. You should run with many different array sizes (they must be sufficiently large to make parallel sorting worthwhile, obviously) and different cutoff schemes.

For varying the number of threads available, you might want to consult the following resources:

* [https://www.callicoder.com/java-8-completablefuture-tutorial/#a-note-about-executor-and-thread-pool (Links to an external site.)](https://www.callicoder.com/java-8-completablefuture-tutorial/#a-note-about-executor-and-thread-pool)
* [https://stackoverflow.com/questions/36569775/how-to-set-forkjoinpool-with-the-desired-number-of-worker-threads-in-completable (Links to an external site.)](https://stackoverflow.com/questions/36569775/how-to-set-forkjoinpool-with-the-desired-number-of-worker-threads-in-completable)

Good luck and enjoy.

**Output for different array sizes and cut off values**

With array size “200000”

Degree of parallelism: 11

cutoff：510000 10times Time:1139ms

cutoff：520000 10times Time:819ms

cutoff：530000 10times Time:779ms

cutoff：540000 10times Time:818ms

cutoff：550000 10times Time:819ms

cutoff：560000 10times Time:812ms

cutoff：570000 10times Time:784ms

cutoff：580000 10times Time:794ms

cutoff：590000 10times Time:784ms

cutoff：600000 10times Time:788ms

cutoff：610000 10times Time:851ms

cutoff：620000 10times Time:791ms

cutoff：630000 10times Time:788ms

cutoff：640000 10times Time:814ms

cutoff：650000 10times Time:811ms

cutoff：660000 10times Time:809ms

cutoff：670000 10times Time:794ms

cutoff：680000 10times Time:806ms

cutoff：690000 10times Time:808ms

cutoff：700000 10times Time:807ms

cutoff：710000 10times Time:800ms

cutoff：720000 10times Time:806ms

cutoff：730000 10times Time:801ms

cutoff：740000 10times Time:819ms

cutoff：750000 10times Time:816ms

cutoff：760000 10times Time:815ms

cutoff：770000 10times Time:935ms

cutoff：780000 10times Time:881ms

cutoff：790000 10times Time:832ms

cutoff：800000 10times Time:902ms

cutoff：810000 10times Time:885ms

cutoff：820000 10times Time:807ms

cutoff：830000 10times Time:794ms

cutoff：840000 10times Time:800ms

cutoff：850000 10times Time:814ms

cutoff：860000 10times Time:837ms

cutoff：870000 10times Time:804ms

cutoff：880000 10times Time:812ms

cutoff：890000 10times Time:824ms

cutoff：900000 10times Time:823ms

cutoff：910000 10times Time:826ms

cutoff：920000 10times Time:814ms

cutoff：930000 10times Time:815ms

cutoff：940000 10times Time:818ms

cutoff：950000 10times Time:827ms

cutoff：960000 10times Time:829ms

cutoff：970000 10times Time:861ms

cutoff：980000 10times Time:850ms

cutoff：990000 10times Time:835ms

cutoff：1000000 10times Time:827ms

With array size “300000”

Degree of parallelism: 11

cutoff：510000 10times Time:1751ms

cutoff：520000 10times Time:1368ms

cutoff：530000 10times Time:1562ms

cutoff：540000 10times Time:1221ms

cutoff：550000 10times Time:1259ms

cutoff：560000 10times Time:1286ms

cutoff：570000 10times Time:1270ms

cutoff：580000 10times Time:1295ms

cutoff：590000 10times Time:1205ms

cutoff：600000 10times Time:1265ms

cutoff：610000 10times Time:1205ms

cutoff：620000 10times Time:1295ms

cutoff：630000 10times Time:1227ms

cutoff：640000 10times Time:1264ms

cutoff：650000 10times Time:1232ms

cutoff：660000 10times Time:1250ms

cutoff：670000 10times Time:1282ms

cutoff：680000 10times Time:1277ms

cutoff：690000 10times Time:1284ms

cutoff：700000 10times Time:1296ms

cutoff：710000 10times Time:1293ms

cutoff：720000 10times Time:1294ms

cutoff：730000 10times Time:1306ms

cutoff：740000 10times Time:1284ms

cutoff：750000 10times Time:1295ms

cutoff：760000 10times Time:1243ms

cutoff：770000 10times Time:1269ms

cutoff：780000 10times Time:1252ms

cutoff：790000 10times Time:1251ms

cutoff：800000 10times Time:1268ms

cutoff：810000 10times Time:1265ms

cutoff：820000 10times Time:1271ms

cutoff：830000 10times Time:1260ms

cutoff：840000 10times Time:1246ms

cutoff：850000 10times Time:1272ms

cutoff：860000 10times Time:1320ms

cutoff：870000 10times Time:1295ms

cutoff：880000 10times Time:1274ms

cutoff：890000 10times Time:1249ms

cutoff：900000 10times Time:1258ms

cutoff：910000 10times Time:1310ms

cutoff：920000 10times Time:1285ms

cutoff：930000 10times Time:1291ms

cutoff：940000 10times Time:1343ms

cutoff：950000 10times Time:1328ms

cutoff：960000 10times Time:1323ms

cutoff：970000 10times Time:1285ms

cutoff：980000 10times Time:1269ms

cutoff：990000 10times Time:1240ms

cutoff：1000000 10times Time:1261ms

With array size “400000”

Degree of parallelism: 11

cutoff：510000 10times Time:2004ms

cutoff：520000 10times Time:1741ms

cutoff：530000 10times Time:1658ms

cutoff：540000 10times Time:1685ms

cutoff：550000 10times Time:1762ms

cutoff：560000 10times Time:1747ms

cutoff：570000 10times Time:1694ms

cutoff：580000 10times Time:1661ms

cutoff：590000 10times Time:1735ms

cutoff：600000 10times Time:1767ms

cutoff：610000 10times Time:1721ms

cutoff：620000 10times Time:1675ms

cutoff：630000 10times Time:1680ms

cutoff：640000 10times Time:1699ms

cutoff：650000 10times Time:1677ms

cutoff：660000 10times Time:1696ms

cutoff：670000 10times Time:1687ms

cutoff：680000 10times Time:1692ms

cutoff：690000 10times Time:1684ms

cutoff：700000 10times Time:1755ms

cutoff：710000 10times Time:1735ms

cutoff：720000 10times Time:1756ms

cutoff：730000 10times Time:1751ms

cutoff：740000 10times Time:1744ms

cutoff：750000 10times Time:1789ms

cutoff：760000 10times Time:1838ms

cutoff：770000 10times Time:1834ms

cutoff：780000 10times Time:1792ms

cutoff：790000 10times Time:1809ms

cutoff：800000 10times Time:1780ms

cutoff：810000 10times Time:1767ms

cutoff：820000 10times Time:1782ms

cutoff：830000 10times Time:1786ms

cutoff：840000 10times Time:1805ms

cutoff：850000 10times Time:1770ms

cutoff：860000 10times Time:1779ms

cutoff：870000 10times Time:1790ms

cutoff：880000 10times Time:1779ms

cutoff：890000 10times Time:1778ms

cutoff：900000 10times Time:1761ms

cutoff：910000 10times Time:1763ms

cutoff：920000 10times Time:1762ms

cutoff：930000 10times Time:1789ms

cutoff：940000 10times Time:1788ms

cutoff：950000 10times Time:1787ms

cutoff：960000 10times Time:1782ms

cutoff：970000 10times Time:1771ms

cutoff：980000 10times Time:1770ms

cutoff：990000 10times Time:1819ms

cutoff：1000000 10times Time:1815ms

**Relationship conclusion**

|  |  |  |  |
| --- | --- | --- | --- |
| Array Size | Range of average execution time for sorting | Range width | Approx Range Width |
| 200000 | 779ms-1139ms | 360 | 350 |
| 300000 | 1205ms-1526ms | 321 | 350 |
| 400000 | 1658ms-2004ms | 352 | 350 |

Relationship conclusion for two different schemes:

1. **Different array sizes:**

As the length of the array increases the range of average execution increases but range width for average execution time for sorting remains approximately the same.

1. **For different cutoff values:**

For different cut off values the average execution time for sorting varies randomly.

**Evidence to support the relationship**

Array size: 200000

|  |  |
| --- | --- |
| CUTOFF | TIME |
| 510000 | 1139 |
| 520000 | 819 |
| 530000 | 779 |
| 540000 | 818 |
| 550000 | 819 |
| 560000 | 812 |
| 570000 | 784 |
| 580000 | 794 |
| 590000 | 784 |
| 600000 | 788 |
| 610000 | 851 |
| 620000 | 791 |
| 630000 | 788 |
| 640000 | 814 |
| 650000 | 811 |
| 660000 | 809 |
| 670000 | 794 |
| 680000 | 806 |
| 690000 | 808 |
| 700000 | 807 |
| 710000 | 800 |
| 720000 | 806 |
| 730000 | 801 |
| 740000 | 819 |
| 750000 | 816 |
| 760000 | 815 |
| 770000 | 935 |
| 780000 | 881 |
| 790000 | 832 |
| 800000 | 902 |
| 810000 | 885 |
| 820000 | 807 |
| 830000 | 794 |
| 840000 | 800 |
| 850000 | 814 |
| 860000 | 837 |
| 870000 | 804 |
| 880000 | 812 |
| 890000 | 824 |
| 900000 | 823 |
| 910000 | 826 |
| 920000 | 814 |
| 930000 | 815 |
| 940000 | 818 |
| 950000 | 827 |
| 960000 | 829 |
| 970000 | 861 |
| 980000 | 850 |
| 990000 | 835 |
| 1000000 | 827 |

Array size 300000

|  |  |
| --- | --- |
| CUTOFF | TIME |
| 510000 | 1751 |
| 520000 | 1368 |
| 530000 | 1562 |
| 540000 | 1221 |
| 550000 | 1259 |
| 560000 | 1286 |
| 570000 | 1270 |
| 580000 | 1295 |
| 590000 | 1205 |
| 600000 | 1265 |
| 610000 | 1205 |
| 620000 | 1295 |
| 630000 | 11227 |
| 640000 | 1264 |
| 650000 | 1232 |
| 660000 | 1250 |
| 670000 | 1282 |
| 680000 | 1277 |
| 690000 | 1284 |
| 700000 | 1296 |
| 710000 | 1293 |
| 720000 | 1294 |
| 730000 | 1306 |
| 740000 | 1284 |
| 750000 | 1295 |
| 760000 | 1243 |
| 770000 | 1269 |
| 780000 | 1252 |
| 790000 | 1251 |
| 800000 | 1268 |
| 810000 | 1265 |
| 820000 | 1271 |
| 830000 | 1260 |
| 840000 | 1246 |
| 850000 | 1272 |
| 860000 | 1320 |
| 870000 | 1295 |
| 880000 | 1274 |
| 890000 | 1249 |
| 900000 | 1258 |
| 910000 | 1310 |
| 920000 | 1285 |
| 930000 | 1291 |
| 940000 | 1343 |
| 950000 | 1328 |
| 960000 | 1323 |
| 970000 | 1285 |
| 980000 | 1269 |
| 990000 | 1240 |
| 1000000 | 1261 |

Array Size 400000

|  |  |
| --- | --- |
| CUTOFF | TIME |
| 510000 | 2004 |
| 520000 | 1741 |
| 530000 | 1658 |
| 540000 | 1685 |
| 550000 | 1762 |
| 560000 | 1747 |
| 570000 | 1694 |
| 580000 | 1661 |
| 590000 | 1735 |
| 600000 | 1767 |
| 610000 | 1721 |
| 620000 | 1675 |
| 630000 | 1680 |
| 640000 | 1699 |
| 650000 | 1677 |
| 660000 | 1696 |
| 670000 | 1687 |
| 680000 | 1692 |
| 690000 | 1684 |
| 700000 | 1755 |
| 710000 | 1735 |
| 720000 | 1756 |
| 730000 | 1751 |
| 740000 | 1744 |
| 750000 | 1789 |
| 760000 | 1838 |
| 770000 | 1834 |
| 780000 | 1792 |
| 790000 | 1809 |
| 800000 | 1780 |
| 810000 | 1767 |
| 820000 | 1782 |
| 830000 | 1786 |
| 840000 | 1805 |
| 850000 | 1770 |
| 860000 | 1779 |
| 870000 | 1790 |
| 880000 | 1779 |
| 890000 | 1778 |
| 900000 | 1761 |
| 910000 | 1763 |
| 920000 | 1762 |
| 930000 | 1789 |
| 940000 | 1788 |
| 950000 | 1787 |
| 960000 | 1782 |
| 970000 | 1785 |
| 980000 | 1770 |
| 990000 | 1819 |
| 1000000 | 1815 |