INFO 6205

Program Structures & Algorithms

Fall 2021

Sutiksh Verma

002122052

Assignment No. 5

- **Task**: To implement a parallel sorting algorithm such that each partition of the array is sorted in parallel. Considering two different schemes for deciding whether to sort in parallel
    - Cutoff (defaults to, say, 1000) which will update according to the first argument in the command line when running.
    - Recursion depth or the number of available threads. Using this determination, 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)
- Show the results of experiments and draw a conclusion (or more) about the efficacy of this method of parallelizing sort. The experiments should involve sorting arrays of sufficient size for the parallel sort to make a difference. Run with many different array sizes (they must be sufficiently large to make parallel sorting worthwhile) and different cutoff schemes

- **Output**:

Size of Array: 500000
Degree of parallelism: 2

cutoff：5000	10times Time:1040ms
cutoff：10000	10times Time:844ms
cutoff：15000	10times Time:219ms
cutoff：20000	10times Time:234ms
cutoff：25000	10times Time:203ms
cutoff：30000	10times Time:188ms
cutoff：35000	10times Time:203ms
cutoff：40000	10times Time:219ms
cutoff：45000	10times Time:218ms
cutoff：50000	10times Time:203ms

Degree of parallelism: 4

cutoff：5000	10times Time:297ms
cutoff：10000	10times Time:187ms
cutoff：15000	10times Time:203ms
cutoff：20000	10times Time:188ms

cutoff：25000 10times Time:203ms
cutoff：30000 10times Time:250ms
cutoff：35000 10times Time:172ms
cutoff：40000 10times Time:218ms
cutoff：45000 10times Time:172ms
cutoff：50000 10times Time:188ms
Degree of parallelism: 8
cutoff：5000 10times Time:250ms
cutoff：10000 10times Time:187ms
cutoff：15000 10times Time:172ms
cutoff：20000 10times Time:187ms
cutoff：25000 10times Time:172ms
cutoff：30000 10times Time:188ms
cutoff：35000 10times Time:187ms
cutoff：40000 10times Time:188ms
cutoff：45000 10times Time:187ms
cutoff：50000 10times Time:156ms
Degree of parallelism: 16
cutoff：5000 10times Time:203ms
cutoff：10000 10times Time:203ms
cutoff：15000 10times Time:187ms
cutoff：20000 10times Time:204ms
cutoff：25000 10times Time:187ms
cutoff：30000 10times Time:187ms
cutoff：35000 10times Time:172ms
cutoff：40000 10times Time:203ms
cutoff：45000 10times Time:188ms
cutoff：50000 10times Time:187ms
Degree of parallelism: 32
cutoff：5000 10times Time:203ms
cutoff：10000 10times Time:188ms
cutoff：15000 10times Time:187ms
cutoff：20000 10times Time:188ms
cutoff：25000 10times Time:172ms
cutoff：30000 10times Time:187ms
cutoff：35000 10times Time:187ms
cutoff：40000 10times Time:188ms
cutoff：45000 10times Time:187ms
cutoff：50000 10times Time:172ms
Degree of parallelism: 64
cutoff：5000 10times Time:203ms
cutoff：10000 10times Time:188ms
cutoff：15000 10times Time:187ms
cutoff：20000 10times Time:188ms
cutoff：25000 10times Time:187ms
cutoff：30000 10times Time:219ms
cutoff：35000 10times Time:172ms
cutoff：40000 10times Time:172ms
cutoff：45000 10times Time:203ms
cutoff：50000 10times Time:171ms

Size of Array: 1000000
Degree of parallelism: 2
cutoff：5000 10times Time:1047ms
cutoff：10000 10times Time:485ms
cutoff：15000 10times Time:500ms
cutoff：20000 10times Time:374ms
cutoff：25000 10times Time:344ms

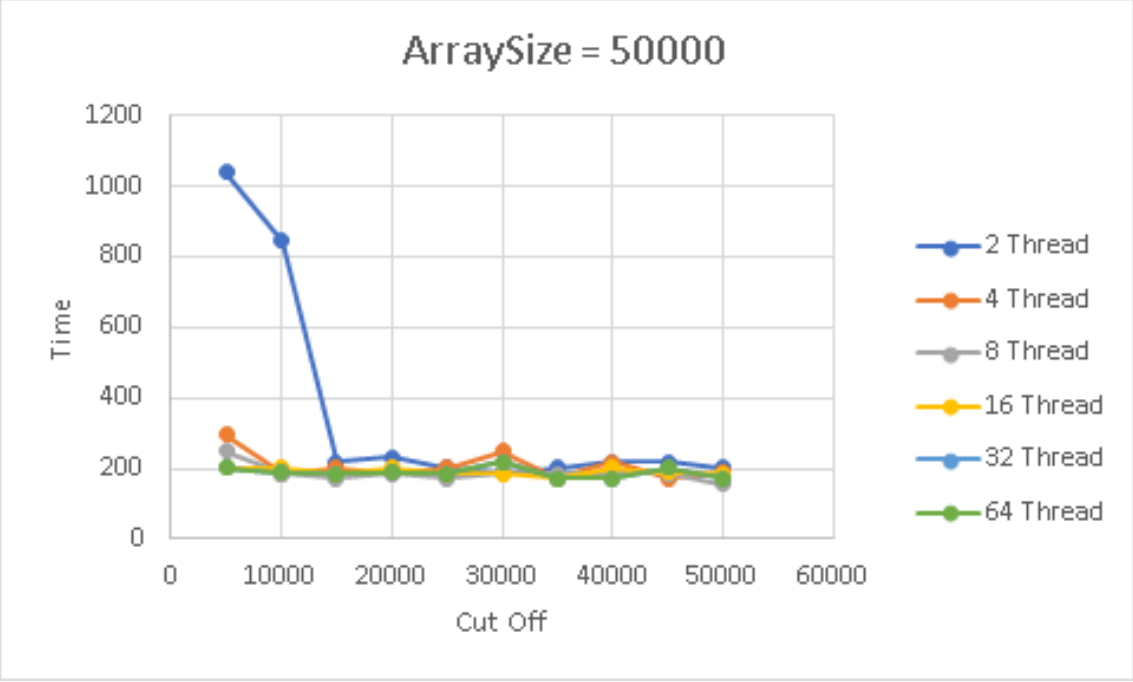
cutoff：30000 10times Time:344ms
cutoff：35000 10times Time:359ms
cutoff：40000 10times Time:344ms
cutoff：45000 10times Time:359ms
cutoff：50000 10times Time:344ms
Degree of parallelism: 4
cutoff：5000 10times Time:437ms
cutoff：10000 10times Time:344ms
cutoff：15000 10times Time:359ms
cutoff：20000 10times Time:328ms
cutoff：25000 10times Time:328ms
cutoff：30000 10times Time:328ms
cutoff：35000 10times Time:328ms
cutoff：40000 10times Time:344ms
cutoff：45000 10times Time:344ms
cutoff：50000 10times Time:328ms
Degree of parallelism: 8
cutoff：5000 10times Time:406ms
cutoff：10000 10times Time:328ms
cutoff：15000 10times Time:328ms
cutoff：20000 10times Time:328ms
cutoff：25000 10times Time:328ms
cutoff：30000 10times Time:312ms
cutoff：35000 10times Time:328ms
cutoff：40000 10times Time:313ms
cutoff：45000 10times Time:312ms
cutoff：50000 10times Time:313ms
Degree of parallelism: 16
cutoff：5000 10times Time:422ms
cutoff：10000 10times Time:328ms
cutoff：15000 10times Time:328ms
cutoff：20000 10times Time:328ms
cutoff：25000 10times Time:328ms
cutoff：30000 10times Time:328ms
cutoff：35000 10times Time:328ms
cutoff：40000 10times Time:313ms
cutoff：45000 10times Time:312ms
cutoff：50000 10times Time:313ms
Degree of parallelism: 32
cutoff：5000 10times Time:390ms
cutoff：10000 10times Time:328ms
cutoff：15000 10times Time:344ms
cutoff：20000 10times Time:312ms
cutoff：25000 10times Time:313ms
cutoff：30000 10times Time:312ms
cutoff：35000 10times Time:313ms
cutoff：40000 10times Time:312ms
cutoff：45000 10times Time:313ms
cutoff：50000 10times Time:328ms
Degree of parallelism: 64
cutoff：5000 10times Time:375ms
cutoff：10000 10times Time:343ms
cutoff：15000 10times Time:328ms
cutoff：20000 10times Time:328ms
cutoff：25000 10times Time:328ms
cutoff：30000 10times Time:313ms
cutoff：35000 10times Time:312ms
cutoff：40000 10times Time:313ms
cutoff：45000 10times Time:296ms
cutoff：50000 10times Time:313ms

Size of Array: 2000000
Degree of parallelism: 2
cutoff:5000 10times Time:1656ms
cutoff:10000 10times Time:1232ms
cutoff:15000 10times Time:844ms
cutoff:20000 10times Time:719ms
cutoff:25000 10times Time:734ms
cutoff:30000 10times Time:766ms
cutoff:35000 10times Time:734ms
cutoff:40000 10times Time:718ms
cutoff:45000 10times Time:719ms
cutoff:50000 10times Time:719ms
Degree of parallelism: 4
cutoff:5000 10times Time:859ms
cutoff:10000 10times Time:719ms
cutoff:15000 10times Time:687ms
cutoff:20000 10times Time:687ms
cutoff:25000 10times Time:672ms
cutoff:30000 10times Time:688ms
cutoff:35000 10times Time:671ms
cutoff:40000 10times Time:688ms
cutoff:45000 10times Time:671ms
cutoff:50000 10times Time:657ms
Degree of parallelism: 8
cutoff:5000 10times Time:812ms
cutoff:10000 10times Time:703ms
cutoff:15000 10times Time:718ms
cutoff:20000 10times Time:657ms
cutoff:25000 10times Time:719ms
cutoff:30000 10times Time:687ms
cutoff:35000 10times Time:656ms
cutoff:40000 10times Time:687ms
cutoff:45000 10times Time:688ms
cutoff:50000 10times Time:703ms
Degree of parallelism: 16
cutoff:5000 10times Time:828ms
cutoff:10000 10times Time:687ms
cutoff:15000 10times Time:703ms
cutoff:20000 10times Time:672ms
cutoff:25000 10times Time:687ms
cutoff:30000 10times Time:672ms
cutoff:35000 10times Time:640ms
cutoff:40000 10times Time:656ms
cutoff:45000 10times Time:672ms
cutoff:50000 10times Time:703ms
Degree of parallelism: 32
cutoff:5000 10times Time:842ms
cutoff:10000 10times Time:703ms
cutoff:15000 10times Time:672ms
cutoff:20000 10times Time:671ms
cutoff:25000 10times Time:672ms
cutoff:30000 10times Time:651ms
cutoff:35000 10times Time:656ms
cutoff:40000 10times Time:641ms
cutoff:45000 10times Time:672ms
cutoff:50000 10times Time:656ms
Degree of parallelism: 64
cutoff:5000 10times Time:828ms

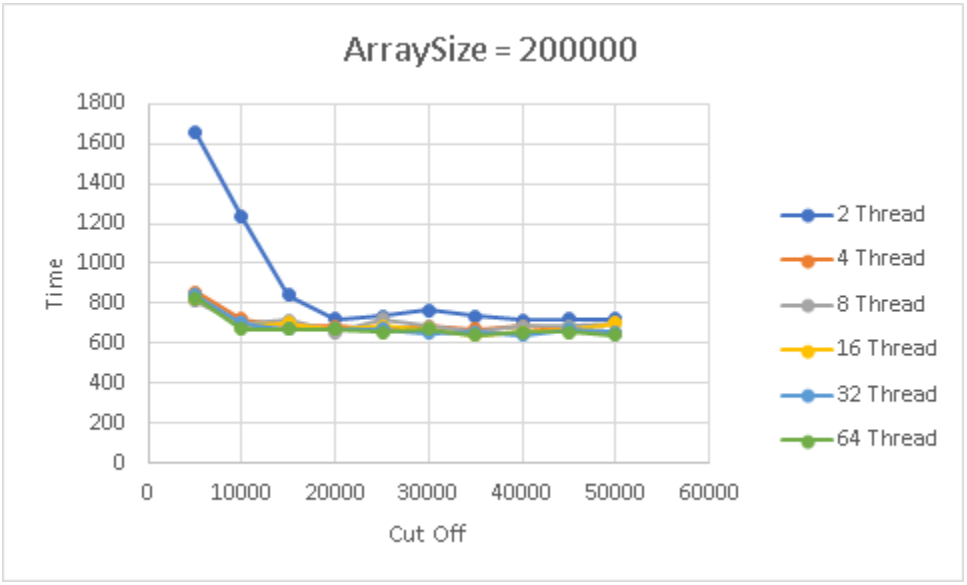
```
cutoff:10000        10times Time:671ms
cutoff:15000        10times Time:672ms
cutoff:20000        10times Time:672ms
cutoff:25000        10times Time:656ms
cutoff:30000        10times Time:672ms
cutoff:35000        10times Time:640ms
cutoff:40000        10times Time:656ms
cutoff:45000        10times Time:656ms
cutoff:50000        10times Time:641ms

Process finished with exit code 0
```

- **Relationship Conclusion**: It can be concluded from the results mentioned above and the graphs that-
    - After changing the cutoff value and the number of threads for different sizes of arrays, the number of threads bigger than 4 does not improve the performance. So, keeping 4 threads is the optimal choice.
    - Referring to the graph, it can be said that for the cutoff value of 25% of the size of the array, the lowest performance time is achieved.
    - Thus, with cutoff value as 25% and number of threads as 4 the optimal performance can be observed

- **Evidence to support Relationship**:




ArraySize = 100000
Time
Cut Off
0
200
400
600
800
1000
1200
0
10000
20000
30000
40000
50000
60000
2 Thread
4 Thread
8 Thread
16 Thread
32 Thread
64 Thread


ArraySize = 200000
Time
Cut Off
0
200
400
600
800
1000
1200
1400
1600
1800
0
10000
20000
30000
40000
50000
60000
2 Thread
4 Thread
8 Thread
16 Thread
32 Thread
64 Thread