Venkata Srinivas Kompally (NUID : 002137855)

Program Structures & Algorithms

Fall 2021

Assignment No. 5

- **Task :**
  - 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.
  - 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.
  - 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).

Output(Terminal)

Size of Array: 500000
Degree of parallelism: 2 The size is: 500000
cutoff : 5000 10times Time:740ms
cutoff : 10000 10times Time:494ms

cutoff：15000 10times Time:436ms
cutoff：20000 10times Time:365ms
cutoff：25000 10times Time:374ms
cutoff：30000 10times Time:392ms
cutoff：35000 10times Time:355ms
cutoff：40000 10times Time:376ms
cutoff：45000 10times Time:347ms
cutoff：50000 10times Time:353ms
Degree of parallelism: 4 The size is: 500000
cutoff：5000 10times Time:421ms
cutoff：10000 10times Time:341ms
cutoff：15000 10times Time:370ms
cutoff：20000 10times Time:345ms
cutoff：25000 10times Time:333ms
cutoff：30000 10times Time:327ms
cutoff：35000 10times Time:323ms
cutoff：40000 10times Time:331ms
cutoff：45000 10times Time:364ms
cutoff：50000 10times Time:307ms
Degree of parallelism: 8 The size is: 500000
cutoff：5000 10times Time:390ms
cutoff：10000 10times Time:330ms
cutoff：15000 10times Time:315ms
cutoff：20000 10times Time:297ms
cutoff：25000 10times Time:289ms
cutoff：30000 10times Time:300ms
cutoff：35000 10times Time:289ms
cutoff：40000 10times Time:339ms
cutoff：45000 10times Time:354ms
cutoff：50000 10times Time:297ms
Degree of parallelism: 16 The size is: 500000
cutoff：5000 10times Time:930ms
cutoff：10000 10times Time:648ms
cutoff：15000 10times Time:633ms
cutoff：20000 10times Time:530ms
cutoff：25000 10times Time:575ms
cutoff：30000 10times Time:561ms
cutoff：35000 10times Time:369ms

cutoff：40000 10times Time:378ms
cutoff：45000 10times Time:352ms
cutoff：50000 10times Time:313ms
Degree of parallelism: 32 The size is: 500000
cutoff：5000 10times Time:3365ms
cutoff：10000 10times Time:1056ms
cutoff：15000 10times Time:1032ms
cutoff：20000 10times Time:579ms
cutoff：25000 10times Time:663ms
cutoff：30000 10times Time:639ms
cutoff：35000 10times Time:242ms
cutoff：40000 10times Time:218ms
cutoff：45000 10times Time:222ms
cutoff：50000 10times Time:251ms
Degree of parallelism: 64 The size is: 500000
cutoff：5000 10times Time:14563ms
cutoff：10000 10times Time:1120ms
cutoff：15000 10times Time:1208ms
cutoff：20000 10times Time:325ms
cutoff：25000 10times Time:269ms
cutoff：30000 10times Time:343ms
cutoff：35000 10times Time:217ms
cutoff：40000 10times Time:199ms
cutoff：45000 10times Time:201ms
cutoff：50000 10times Time:199ms

Degree of parallelism: 2 The size is: 1000000
cutoff：5000 10times Time:974ms
cutoff：10000 10times Time:825ms
cutoff：15000 10times Time:772ms
cutoff：20000 10times Time:672ms
cutoff：25000 10times Time:668ms
cutoff：30000 10times Time:657ms
cutoff：35000 10times Time:673ms
cutoff：40000 10times Time:667ms
cutoff：45000 10times Time:647ms
cutoff：50000 10times Time:639ms
Degree of parallelism: 4 The size is: 1000000

cutoff：5000 10times Time:831ms
cutoff：10000 10times Time:719ms
cutoff：15000 10times Time:707ms
cutoff：20000 10times Time:646ms
cutoff：25000 10times Time:715ms
cutoff：30000 10times Time:692ms
cutoff：35000 10times Time:677ms
cutoff：40000 10times Time:646ms
cutoff：45000 10times Time:637ms
cutoff：50000 10times Time:635ms
Degree of parallelism: 8 The size is: 1000000
cutoff：5000 10times Time:794ms
cutoff：10000 10times Time:723ms
cutoff：15000 10times Time:685ms
cutoff：20000 10times Time:688ms
cutoff：25000 10times Time:648ms
cutoff：30000 10times Time:654ms
cutoff：35000 10times Time:651ms
cutoff：40000 10times Time:644ms
cutoff：45000 10times Time:642ms
cutoff：50000 10times Time:635ms
Degree of parallelism: 16 The size is: 1000000
cutoff：5000 10times Time:1289ms
cutoff：10000 10times Time:1125ms
cutoff：15000 10times Time:1112ms
cutoff：20000 10times Time:1044ms
cutoff：25000 10times Time:993ms
cutoff：30000 10times Time:1010ms
cutoff：35000 10times Time:958ms
cutoff：40000 10times Time:936ms
cutoff：45000 10times Time:1000ms
cutoff：50000 10times Time:1005ms
Degree of parallelism: 32 The size is: 1000000
cutoff：5000 10times Time:3863ms
cutoff：10000 10times Time:1741ms
cutoff：15000 10times Time:1663ms
cutoff：20000 10times Time:1444ms
cutoff：25000 10times Time:1479ms

cutoff：30000 10times Time:1468ms
cutoff：35000 10times Time:934ms
cutoff：40000 10times Time:995ms
cutoff：45000 10times Time:897ms
cutoff：50000 10times Time:944ms
Degree of parallelism: 64 The size is: 1000000
cutoff：5000 10times Time:16086ms
cutoff：10000 10times Time:2500ms
cutoff：15000 10times Time:2567ms
cutoff：20000 10times Time:1311ms
cutoff：25000 10times Time:1273ms
cutoff：30000 10times Time:1371ms
cutoff：35000 10times Time:600ms
cutoff：40000 10times Time:592ms
cutoff：45000 10times Time:639ms
cutoff：50000 10times Time:582ms

Degree of parallelism: 2 The size is: 2000000
cutoff：5000 10times Time:1899ms
cutoff：10000 10times Time:1782ms
cutoff：15000 10times Time:1524ms
cutoff：20000 10times Time:1899ms
cutoff：25000 10times Time:1826ms
cutoff：30000 10times Time:1583ms
cutoff：35000 10times Time:1798ms
cutoff：40000 10times Time:1691ms
cutoff：45000 10times Time:1574ms
cutoff：50000 10times Time:1436ms
Degree of parallelism: 4 The size is: 2000000
cutoff：5000 10times Time:1868ms
cutoff：10000 10times Time:1663ms
cutoff：15000 10times Time:2234ms
cutoff：20000 10times Time:2822ms
cutoff：25000 10times Time:1473ms
cutoff：30000 10times Time:1953ms
cutoff：35000 10times Time:1377ms
cutoff：40000 10times Time:1346ms

cutoff：45000 10times Time:1355ms
cutoff：50000 10times Time:1338ms
Degree of parallelism: 8 The size is: 2000000
cutoff：5000 10times Time:1894ms
cutoff：10000 10times Time:1530ms
cutoff：15000 10times Time:1715ms
cutoff：20000 10times Time:1841ms
cutoff：25000 10times Time:1515ms
cutoff：30000 10times Time:1488ms
cutoff：35000 10times Time:1376ms
cutoff：40000 10times Time:1451ms
cutoff：45000 10times Time:1440ms
cutoff：50000 10times Time:1481ms
Degree of parallelism: 16 The size is: 2000000
cutoff：5000 10times Time:2472ms
cutoff：10000 10times Time:1856ms
cutoff：15000 10times Time:1897ms
cutoff：20000 10times Time:1763ms
cutoff：25000 10times Time:1802ms
cutoff：30000 10times Time:1906ms
cutoff：35000 10times Time:1731ms
cutoff：40000 10times Time:1835ms
cutoff：45000 10times Time:1852ms
cutoff：50000 10times Time:2030ms
Degree of parallelism: 32 The size is: 2000000
cutoff：5000 10times Time:4819ms
cutoff：10000 10times Time:2859ms
cutoff：15000 10times Time:4307ms
cutoff：20000 10times Time:3176ms
cutoff：25000 10times Time:2270ms
cutoff：30000 10times Time:2518ms
cutoff：35000 10times Time:1876ms
cutoff：40000 10times Time:2039ms
cutoff：45000 10times Time:2506ms
cutoff：50000 10times Time:2481ms
Degree of parallelism: 64 The size is: 2000000
cutoff：5000 10times Time:23287ms
cutoff：10000 10times Time:3638ms

cutoff : 15000 10times Time:3409ms
cutoff : 20000 10times Time:3078ms
cutoff : 25000 10times Time:3025ms
cutoff : 30000 10times Time:2991ms
cutoff : 35000 10times Time:1881ms
cutoff : 40000 10times Time:1970ms
cutoff : 45000 10times Time:1963ms
cutoff : 50000 10times Time:2084ms

Process finished with exit code 0

- **Relationship Conclusion:**

➔ The experiment was done on Dell Inspiron Laptop - Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz 1.99 GHz.

I took array sizes as 500000, 1000000, 2000000.

And thread sizes as 2, 4, 8, 16, 32 and 64(powers of 2) for testing

➔ From the below scatter plots, we were able to come to conclusion that size required is 16 and above that, there is no significant use and will not help in reducing the time complexity.

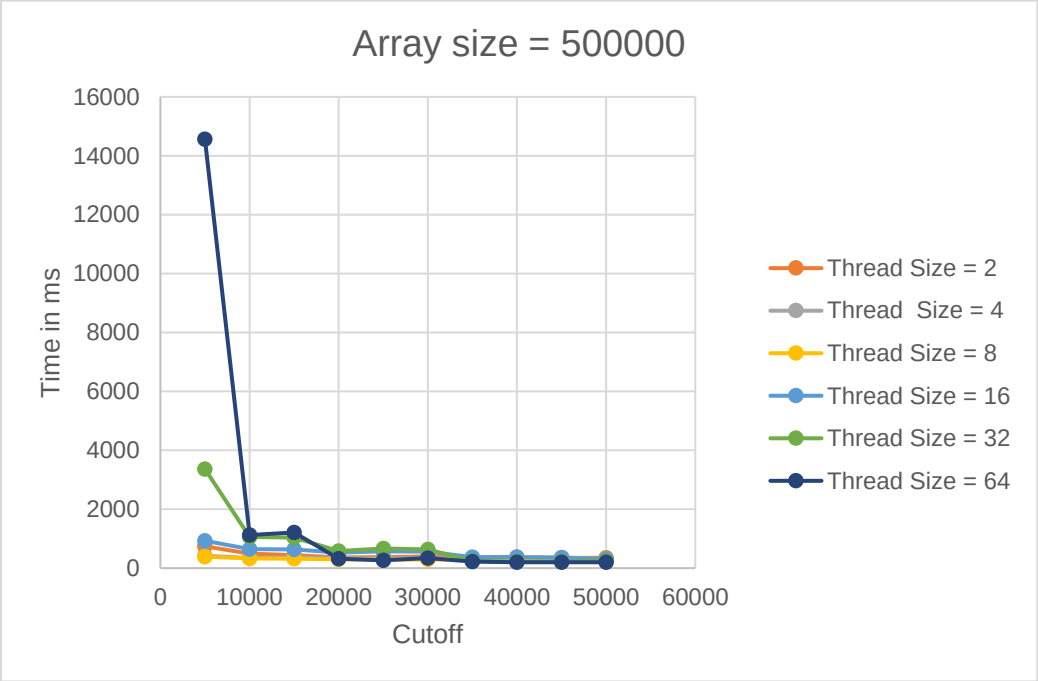
➔ We can also observe that 25% of the array size is best to achieve the best time for performance of the program.

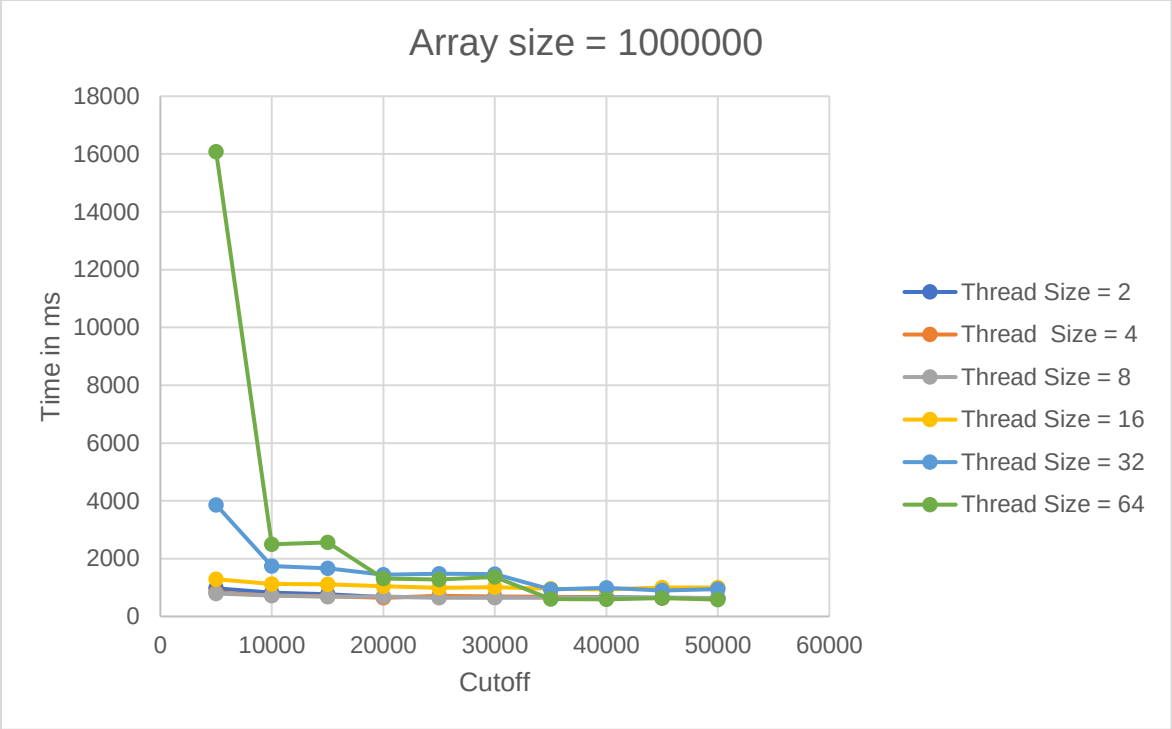
➔ $t = 2\text{^}d$ (t = number of threads available, d = Recursion Depth)

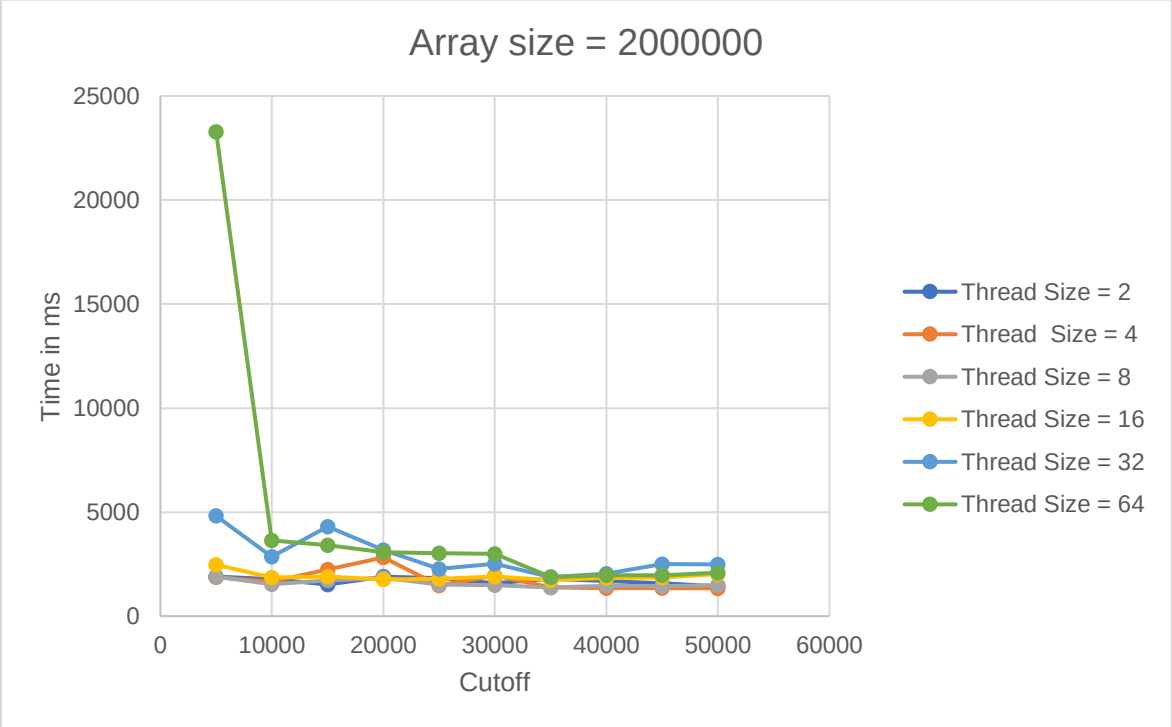
➔ Maximum Depth Possible is : lg(arrsize/cutoff)

The folder for all the data is src/assignment5_results

Graphical Representation - Evidence to support the Relationship




Array size = 1000000
Time in ms
18000
16000
14000
12000
10000
8000
6000
4000
2000
0
0
10000
20000
30000
40000
50000
60000
Cutoff
Thread Size = 2
Thread Size = 4
Thread Size = 8
Thread Size = 16
Thread Size = 32
Thread Size = 64


Array size = 2000000
Time in ms
25000
20000
15000
10000
5000
0
0
10000
20000
30000
40000
50000
60000
Cutoff
Thread Size = 2
Thread Size = 4
Thread Size = 8
Thread Size = 16
Thread Size = 32
Thread Size = 64