Ren Junyan (NUID: 001529948)

INFO 6205

Program Structures & Algorithms

Fall 2020

Assignment No5

1. Task:

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.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.

2. Output:

```
D:\Java\jdk\bin\java.exe "-javaagent:D:\intel
Degree of parallelism: 7
the size of array is: 1000000
cutoff: 510000      10times Time:1318ms
cutoff: 520000      10times Time:705ms
cutoff: 530000      10times Time:690ms
cutoff: 540000      10times Time:667ms
cutoff: 550000      10times Time:663ms
cutoff: 560000      10times Time:656ms
cutoff: 570000      10times Time:669ms
cutoff: 580000      10times Time:684ms
cutoff: 590000      10times Time:662ms
cutoff: 600000      10times Time:657ms
Average time: 737.0ms

Degree of parallelism: 7
the size of array is: 2000000
cutoff: 510000      10times Time:1056ms
cutoff: 520000      10times Time:1070ms
cutoff: 530000      10times Time:1161ms
cutoff: 540000      10times Time:1128ms
cutoff: 550000      10times Time:1145ms
cutoff: 560000      10times Time:1154ms
cutoff: 570000      10times Time:1077ms
cutoff: 580000      10times Time:1033ms
cutoff: 590000      10times Time:1027ms
cutoff: 600000      10times Time:1238ms
Average time: 1108.0ms

the size of array is: 4000000
cutoff: 510000      10times Time:2140ms
cutoff: 520000      10times Time:2132ms
cutoff: 530000      10times Time:2178ms
cutoff: 540000      10times Time:2174ms
cutoff: 550000      10times Time:2166ms
cutoff: 560000      10times Time:2169ms
cutoff: 570000      10times Time:2155ms
cutoff: 580000      10times Time:2169ms
cutoff: 590000      10times Time:2248ms
cutoff: 600000      10times Time:2126ms
Average time: 2165.0ms

Degree of parallelism: 7
the size of array is: 8000000
cutoff: 510000      10times Time:4811ms
cutoff: 520000      10times Time:4527ms
cutoff: 530000      10times Time:4889ms
cutoff: 540000      10times Time:4693ms
cutoff: 550000      10times Time:4403ms
cutoff: 560000      10times Time:4436ms
cutoff: 570000      10times Time:4481ms
cutoff: 580000      10times Time:4702ms
cutoff: 590000      10times Time:4918ms
cutoff: 600000      10times Time:4930ms
Average time: 4679.0ms

the size of array is: 16000000
cutoff: 510000      10times Time:9890ms
cutoff: 520000      10times Time:9814ms
cutoff: 530000      10times Time:10230ms
cutoff: 540000      10times Time:10515ms
cutoff: 550000      10times Time:9628ms
cutoff: 560000      10times Time:9585ms
cutoff: 570000      10times Time:9527ms
cutoff: 580000      10times Time:9624ms
cutoff: 590000      10times Time:9608ms
cutoff: 600000      10times Time:9492ms
Average time: 9791.0ms

Degree of parallelism: 7
the size of array is: 32000000
cutoff: 510000      10times Time:21280ms
cutoff: 520000      10times Time:21392ms
cutoff: 530000      10times Time:21100ms
cutoff: 540000      10times Time:21821ms
cutoff: 550000      10times Time:21490ms
cutoff: 560000      10times Time:21529ms
cutoff: 570000      10times Time:21799ms
cutoff: 580000      10times Time:21634ms
cutoff: 590000      10times Time:21688ms
cutoff: 600000      10times Time:21232ms
Average time: 21496.0ms
```

3. Relationship conclusion:

For parallel sort, the value of cutoff is related to the times of recursion, affecting the number of divided partitions, but it seems to be independent of the time of implementation. On the other hand, the time of parallel sort grows gradually quickly with the surge of size of array based on the same value of cutoff showed clearly on the graph, that is obvious result.

4. Evidence to support relationship: