COMP 428: Parallel Programming

Winter 2012 Assignment 1

Grégoire Morpain - #6398391

gregoire.morpain@gmail.com

Q.1.

1. Cf. Source Code
2. All tests have been realized with 5000000Darts on 1000Rounds.

2 workers ~> 28.767s

4 workers ~> 17.618s

6 workers ~> 14.066s

8 workers ~> 12.607s

10 workers ~> 13.145s

14 workers ~> 14.611s

NB. Tests realized on an 8-cores processor.

1. The code is not 100% implemented therefore I have no relevant data for this experiment.

d)

The speedup decreases if we our number of workers is superior to our processor’s real number of cores.

Q.2.

Master pseudo-code

create X workers

send a task to each workers

while workToDo {

receive result from worker

schedule next task

send task to the worker

}

while workToDo {

receive task from master

execute task

send result to master

}

Worker main loop pseudo-code

Q.3.

a) Yes it is close to possible, for example if we apply a divide-and-conquer parallelization strategy to a merge sort, the number of comparisons is close to theoretical minimum for comparison-based sorting:

log n! ≈ n lg n - 1.44 n

b) This statement is false. With fine-grained granularity the data is transferred among processors frequently in amount of few memory words; the finer the granularity, the greater the potential of speed-up, but it increases the overheads of synchronization.

To get the best parallel performance, you have to find the right amount of granularity; otherwise you can suffer from increased communication overhead or on the other side from load imbalance.

Q.4.