

Assignment 3
COMP 409
Marco Guida
260803123

Q1:

$n = 200000$

$s = 50$

For $t = 1$

2228ms

For $t = 2$

934ms

Speedup = 2.385

For $t = 5$

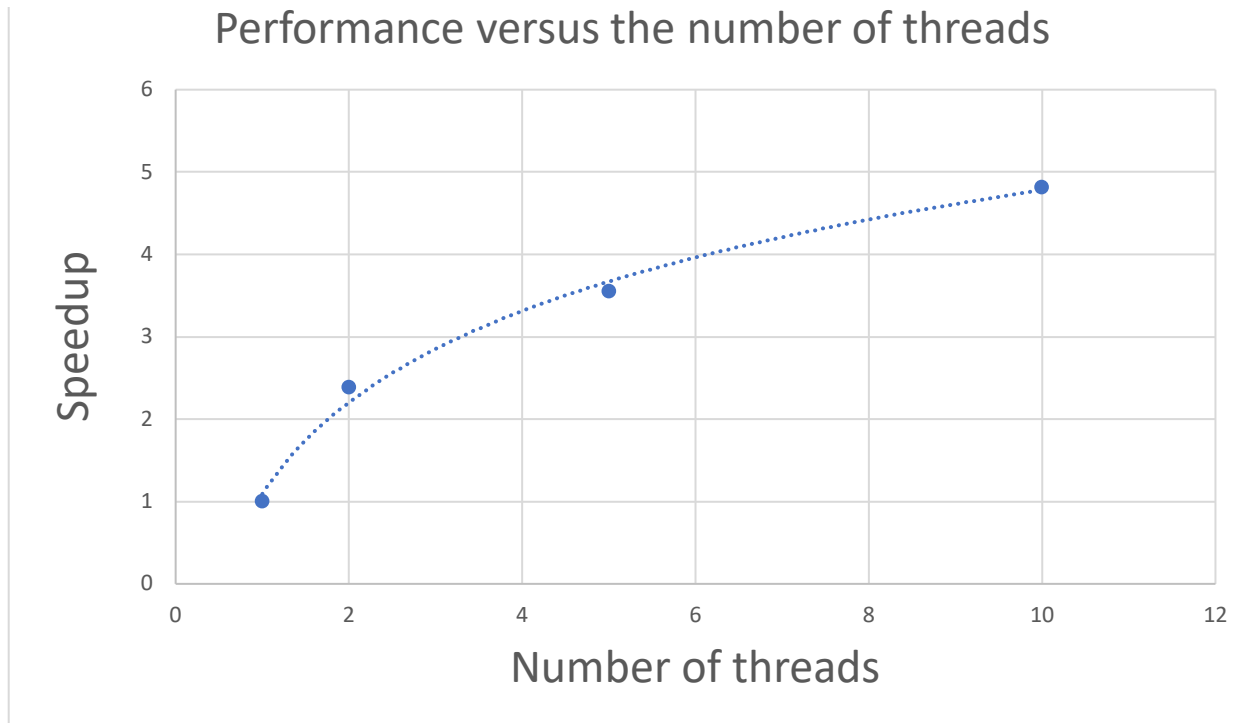
628ms

Speedup = 3.547

For $t = 10$

463ms

Speedup = 4.812



I observed a speedup with each additional thread. As expected, the magnitude of the speedup increase gets smaller with each additional thread as speedup is limited by that part of the program that cannot be parallelized. For this program, the serial part that cannot be parallelized is the sequential combination of triples after each thread computed the resulting triple for their designated substring. The speedup is especially noticeable for strings of longer lengths (like 200000 characters long) as single-threaded programs will suffer in combining the many base case triples (base case: one triple for each character).

Q2:

d = 20
n = 1000
t = 2

(4, 1) = 11816ms
(4, 2) = 11708ms
(4, 4) = 11756ms

(8, 1) = 11643ms
(8, 2) = 11527ms
(8, 4) = 11511ms

(16, 1) = 11402ms

(16, 2) = 11535ms

(16, 4) = 11243ms

No, I don't think the elimination array is being used effectively, I only observe 200-400ms differences between 1 and 4 entries in the array, regardless of how many threads are being used. I think the rather small timeout window of 2ms for the elimination array makes it more difficult for threads to attempt exchanges.

However, I also know that my code contains a bug I cannot find the solution to with regards to the reuse (re-pushing) of AtomicStampedReference nodes (hence the 1000 remaining node count during some executions of my code). Perhaps this is related to why the results aren't as expected.