Lab 6 - Parallelizing techniques

January 4, 2019

1 Goal

The goal of this lab is to implement a simple but non-trivial parallel algorithm.

2 Requirement

Perform the multiplication of 2 polynomials. Use both the regular O(n^2) algorithm and the Karatsuba algorithm, and each in both the sequential form and a parallelized form. Compare the 4 variants.

3 Computer Specification

- *CPU:* Intel Core i7-7500U, 2.90GHz
- *RAM:* 8 GB
- *System type:* 64-bit

4 Short Description of the Implementation

Algorithm - Backtracking:

- *Regular polynomial multiplication*
- *Karatsuba algorithm*

Parallelization - Used Java's ThreadPool combined with custom Runnable objects to achieve parallelization

4.1 Regular polynomial multiplication

- *Complexity:* $O(n^2)$
- *Step 1: Distribute each term of the first polynomial to every term of the second polynomial. Remember that when you multiply two terms together you must multiply the coefficient (numbers) and add the exponents*
- *Step 2: Combine like terms (if you can)*

4.2 Karatsuba algorithm

- *Complexity:* $O(n^{\log_2 3})$
- *A fast multiplication algorithm that uses a divide and conquer approach to multiply two numbers.*

5 Performance Tests

note: by level 'x' i am referring that the algorithms were used to multiply 2 polynomials of rank x * 100, with coefficients being random numbers of x * 10 digits.

Algorithm	Level 1	Level 5	Level 8	Level 20
regular sequential	72 ms	348 ms	1199 ms	5995 ms
regular parallelized	28 ms	209 ms	724 ms	6383 ms
karatsuba sequential	29 ms	190 ms	507 ms	5981 ms
karatsuba parallelized	17 ms	238 ms	679 ms	5079 ms

Thought all the tests I've put those algorithms to, the results were for the most part as expected, although the results may vary quite a bit (see that up to level 8 the parallelized version of the regular algorithm has the lead, but suddenly at level 20 it takes quite a bit more time than the sequential one). There are a lot of factors that can be responsible for those inconsistencies, like background processes, memory usage, and the implementation itself.

6 Conclusion

- *For the most part, the parallelized versions of the algorithms run faster.*
- *Karatsuba's is clearly superior to the regular algorithm and for large numbers it would be preferred*