Leonard Mihalcea

Lab 5 - Parallelizing techniques

Goal

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

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.

The documentation will describe:

- the algorithms,
- the synchronization used in the parallelized variants,
- the performance measurements

Bonus: do the same for big numbers.

Computer Specification

CPU: Intel Core i5-7300HQ
RAM: 16GB 2

Implementation

1. Regular polynomial multiplication - Naive Algorithm - Single Threaded
2. Regular polynomial multiplication - Naive Algorithm - Multi Threaded
3. Karatsuba's Algorithm - Naive Algorithm - Single Threaded
4. Karatsuba's Algorithm - Naive Algorithm - Multi Threaded

1. Regular polynomial multiplication - Naive Algorithm - Single Threaded

Complexity: $O(n^2)$

We distribute each term of the first polynomial to every term of the second polynomial. When we multiply two terms together we must multiply the coefficient and add the exponents.
After computing that, we must combine the terms to get the final result

2. Regular polynomial multiplication - Naive Algorithm - Multi Threaded

Complexity: $O(n^2)$

We distribute each term of the first polynomial to every term of the second polynomial. When we multiply two terms together we must multiply the coefficient and add the exponents.
After computing that, we must combine the terms to get the final result

3. Karatsuba's Algorithm - Single Threaded

Complexity: $O(n \log_2 3)$

A fast multiplication algorithm that uses a divide and conquer approach to multiply two numbers.

4. Karatsuba's Algorithm - Multi Threaded

Complexity: $O(n \log_2 3)$

A fast multiplication algorithm that uses a divide and conquer approach to multiply two numbers.

Tests

Degree = 10

O(n2) algorithm - sequential
Execution time : 1 ms

O(n2) algorithm - parallelized
Execution time : 31 ms
Karatsuba sequential multiplication of polynomials:
Execution time : 2 ms
Karatsuba sequential multiplication of polynomials:
Execution time : 19 ms

Degree = 100

O(n2) algorithm - sequential
Execution time : 5 ms
O(n2) algorithm - parallelized
Execution time : 7 ms
Karatsuba sequential multiplication of polynomials:
Execution time : 10 ms
Karatsuba sequential multiplication of polynomials:
Execution time : 14 ms

Degree = 1000

O(n2) algorithm - sequential
Execution time : 49 ms
O(n2) algorithm - parallelized
Execution time : 90 ms
Karatsuba sequential multiplication of polynomials:
Execution time : 129 ms
Karatsuba sequential multiplication of polynomials:
Execution time : 135 ms

Degree = 10000

O(n2) algorithm - sequential
Execution time : 2783 ms
O(n2) algorithm - parallelized
Execution time : 869 ms
Karatsuba sequential multiplication of polynomials:
Execution time : 823 ms
Karatsuba sequential multiplication of polynomials:
Execution time : 732 ms

Degree = 100000

O(n2) algorithm - sequential
Execution time : 176005 ms
O(n2) algorithm - parallelized
Execution time : 50037 ms
Karatsuba sequential multiplication of polynomials:
Execution time : 41465 ms
Karatsuba sequential multiplication of polynomials:
Execution time : 35132 ms