Lab 6 - Parallelizing techniques

Hiticas Claudiu Emanuel

Group 924 Year II

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 sequencial 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

3. Computer specifications

MacBook Air (13-inch, Early 2015)

1.6 GHz Dual-Core Intel Core i5

8 GB 1600 MHz DDR3
Intel HD Graphics 6000 1536 MB

4. Short description of the implementation

Algorithms:
  * Regular polynomial multiplication
  * Karatsuba algorithm

4.1 Regular polynomial multiplication

- Complexity: O(n^2)
- Distribute each term of the 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

4.2 Karatsuba algorithm

- Complexity: O(n^log3)
- A fast multiplication algorithm that uses a divide and conquers approach to multiply two numbers

5. Performance test

Generate random 100 numbers from 1 to 1000

```
Karatsuba algorithm on took me 0.00682453 seconds
Karatsuba algorithm (threaded) took me 0.00843777 
seconds
Naive algorithm took me 0.000214652 seconds
Naive algorithm (thread) took me 0.00367946 seconds
```

Generate random 1000 numbers from 1 to 1000

```
Karatsuba algorithm on took me 0.152756 seconds
Karatsuba algorithm (threaded) took me 0.103809 
seconds
Naive algorithm took me 0.0127782 seconds
Naive algorithm (thread) took me 0.0414323 seconds
```

Generate random 10000 numbers from 1 to 10000

```
Karatsuba algorithm on took me 8.88445 seconds
Karatsuba algorithm (threaded) took me 5.62269 
seconds
Naive algorithm took me 1.11308 seconds
Naive algorithm (thread) took me 1.14927 seconds
```