PDP Lab 7 – MPI Polynomial Multiplication

Alex Ovidiu Popa, 936/1

Problem Requirements

Algorithm Descriptions

Regular polynomial multiplication

- Complexity: O(n^2)
- Step 1: Distribute each term of the first polynomial to every term of the second polynomial. When you multiply two terms together you must multiply the coefficient (numbers) and add the exponents.
- Step 2: Sum up the terms resulted from the multiplications which result in the same exponent, the solo ones stay the same.

Karatsuba algorithm

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

Distributing the workload to each node was done by dividing the polynomial length to the number of worker processes (namely MPI.size()-1 because we exclude the master process) and then going step by step with this computation, until the last worker process was reached.

At each iteration i, MPI.send is used to send data to be computed by the worker node i. The node sends in return an incomplete result.

After all the sending was done, the master process receives all the incomplete results and adds them up to the final result.

Computer Specifications

- Intel Core i7-4790 CPU @ 3.60GHz, 4 Cores, 8 Logical Processors
- 16GB RAM

Tests Run

Times are measured in ms

Order	Nodes	Regular MPI	Ktsb MPI
10	4	94	100
50	4	81	104
100	4	85	122
500	4	133	205
1000	4	184	340
5000	4	397	1025
1000	3	150	323
5000	3	473	758
5000	5	427	1300
10000	4	989	1805
100000	4	46903	37670

Conclusions

1. In comparison with lab 5, distributing data to the nodes and creating multiple nodes takes way more time than creating threads or using recursion.
2. As the number of nodes increases, the time increases as well.
3. For small orders (below millions), regular multiplication is preferred over Karatsuba.