

Lab 7 - Distributed polynomial multiplication

Şut George-Mihai

I. ASSIGNMENT TASK

The goal of this lab is to implement a distributed algorithm using MPI. Perform the multiplication of 2 polynomials, by distributing computation across several nodes using MPI. Use both the regular $O(n^2)$ algorithm and the Karatsuba algorithm. Compare the performance with the "regular" CPU implementation from lab 5.

II. IMPLEMENTATIONS

The most important method is *computeMasterMultiplicationTask*, which is called from the master process and sends to each worker process both polynomials and an interval/range for the second polynomial. The coefficients included in this range are multiplied with the first polynomial.

runWorkerComputingRegularMultiplication and *runWorkerComputingKaratsubaMultiplication* are the methods which run the partial computations for each of the two types of multiplication algorithms and the results are sent back to the master process.

The functions used for the process communication are *MPI_Send* and *MPI_Recv*, whose essential parameters are the content to be sent / received (the void pointer parameter) and the destination / source as a number which represents the process ID.

The sequential regular multiplication algorithm is the trivial, normal algorithm for multiplying numbers and polynomials.

For the Karatsuba algorithm, we split each polynomial in half recursively and the number of multiplications is reduced by 1. In a classic example of normal multiplication, 4 multiplications are required, while Karatsuba's algorithm only requires 3 multiplications.

III. RESULTS

The results from Lab 5 are also included in the below table.

The main conclusion is that the distributed regular and Karatsuba algorithms seem to be quite slower for large polynomial sizes compared to the sequential and parallelized versions from Lab-5, because communication between threads is significantly faster than inter-process communication.

Degree	$O(n^2)$ seq.	$O(n^2)$ par.	$O(n^{\log_2 3})$ seq.	$O(n^{\log_2 3})$ par.	$O(n^2)$ distr.	$O(n^{\log_2 3})$ distr.
1023	0.24838	0.5049	0.57698	0.21373	0.38334	0.98020
8191	16.4522	4.1569	15.497	5.88652	26.7514	39.4723

TABLE I: Table with execution times (seconds)