

Homework 5

due Nov 16, 2022

Exercise 5 : Parallelized matrix times vector

Let A be a $(m \times n)$ matrix and x a n -dimensional vector. Write a MPI program which does the multiplication

$$A \cdot x = y.$$

Use a block-row distribution of the matrix A while a copy of the vector x is stored on all the processes.

- Matrix and vectors should be dynamically (and not statically) allocated (hint: `malloc`). For the matrix use a one-dimensional index, i.e. the matrix element $A[i][j]$ should be addressed as $A[i \cdot n + j]$.
- It is fundamental that in the entire program you never allocate a global matrix of size $(m \times n)$.
- Generate (for example with MATLAB) an input file containing m , n , a random matrix A and a random vector x . Run your program with $m = 2048$ and $n = 2001$ changing the number of processes $p = 2, 4, 8, 16$ (*strong scaling*) and hand in a graph of the CPU time as a function of p .
- To verify the correctness of your program, compare the norm of the vector y computed by your program with that computed with the same program with which you generated the input file (for example MATLAB).

Together with your MPI source code and the study of the strong scaling, hand in also all the scripts which you have used to generate the input file and to verify the output.

(15 points)