
COURSE WORKSHOP

"INTRODUCTION TO PARALLEL COMPUTING"

EXERCISE-II 2022-2023 (20%)

You are asked to make an MPI program (in C language), which given one vector X (length n elements $x_i \mid i=0 \dots n-1$), **to compute** in parallel environment 'p' processors and print to the screen (as output) the following (where m , x_{min} , x_{max} are the mean, minimum and maximum element of the vector X respectively):

(a) How many elements of the vector X have a value less than and how many greater than mean value (m) thereof.

(b) The dispersion of the elements of the vector X :

$$var = ((x_0 - m)^2 + (x_1 - m)^2 + (x_2 - m)^2 + \dots + (x_{n-1} - m)^2) / n$$

(c) A new vector D where each element \tilde{y}_i will be equal to its percentage relationship corresponding element (x_i) of the vector X with the difference of the maximum-minimum values of the entire X vector:

$$\tilde{y}_i = ((x_i - x_{min}) / (x_{max} - x_{min})) * 100$$

(d) What is the largest value of the vector \tilde{y} and for which element x_i in particular observed (it should print the position i of the element in the vector, its value element and its \tilde{d}_i).

(e) The vector of prefix sums of the elements of X .

Try to use **only functions** in your implementation as much as possible **collective communication**.

The total computational load required should be equally distributed across 'p' processors of the parallel environment. Also, every processor should receives (holds) in its local memory only the input data it uses for local (own) calculations.

Assume first that 'n' is an integer multiple of 'p'. Then try extend your program so that it behaves correctly for anyone combination of 'n' and 'p' values (using MPI_Scatterv/MPI_Gatherv functions).

Regarding the requests of sub-question (d) (and in particular for the calculation of position) study the example *"mpi_groups_plus.doc"* (in *Documents->AP-Workshop->Basic Examples->Basic-Word-Files.zip* / see especially last page about using MPI_Reduce with maxloc/minloc operators).

Regarding the prompts of sub-question (e) first consult slides 2,4,5 from the theory slide file #9 (non-recursive prefix calculation algorithm) and the example *Documents->AP-Lab->prefix_sums_example.pdf*. For your implementation use the MPI_Scan function (see *scan.c* file in *Documents->AP-Lab->Basic Examples->More_examples.zip*), and where necessary the MPI_Send/MPI_Recv functions.

Your program should work iteratively with a menu of options.

Method and Date of Delivery: _____

The Exercise should be submitted electronically (through the Eclass platform) by on **Sunday 15/1/2023**.

Deliverables: Commented code, documentation, and test runs.