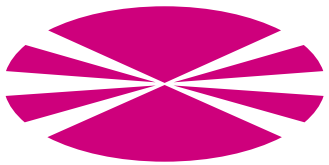
Concurrency and Parallelism. Block II Parallelism

Assignment 2: MPI collectives when counting the occurrences of a letter

Spring 2023



MPI collectives when counting the occurrences of a letter

Parallelization performed in assignment 1 + improvements assignment 2!

- SPMD implementation
- I/O (scanf/printf) is made by process 0
- Distribute n and L to all the processes (with Send/Recv) Now with MPI collective operation!
- Divide the workload of the for loop with "step" i+=numprocs instead of i++
- Gather the number of occurrences detected by each process (with Send/Recv) Now with MPI collective operation!

MPI collectives when counting the occurrences of a letter

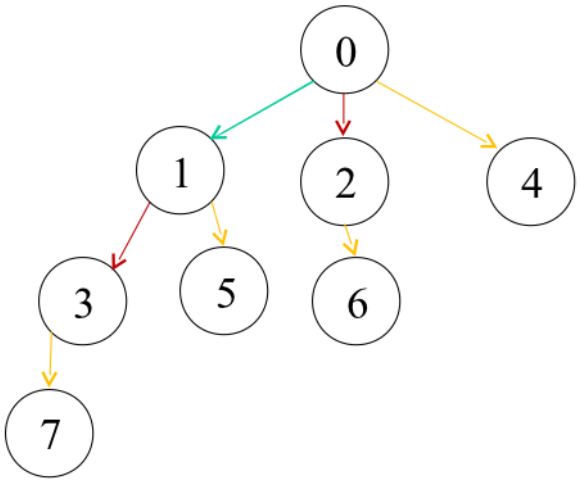
Usage of MPI collectives

- Initially use standard MPI collective operations
- Implement the collective following a binomial tree approach. We will call MPI_BinomialColective to this implementation to use ONLY in the distribution of n and L
- Later introduce own implementation of the collective (with same header as the standard collective) ONLY for the recollection of *count*, initially using the same Send/Recv operations as in the implementation without collectives (for loop with Sends), implementation that we will call MPI_FlattreeColective. Assume that the operation to perform is an addition. The remaining parameters of the function must be the same ones as in the standard MPI collective (even the error checking).

MPI collectives when counting the occurrences of a letter

Implementation of Bcast with binomial tree (MPI_BinomialBcast):

- Same parameters as MPI_Bcast (see man page of MPI_Bcast to obtain the header), assuming for simplicity that the root is 0
- In step "i" the processes with $myrank < 2^{i-1}$ communicate with the process $myrank + 2^{i-1}$



Paso 1: 0→1
Paso 2: 0→2, 1→3
Paso 3: 0→4, 1→5, 2→6, 3→7

Conditions of the assignment

- Assigned points: 0.5
- Deadline: 17-21 April
- It must be done in couples
- Defended in the laboratory class