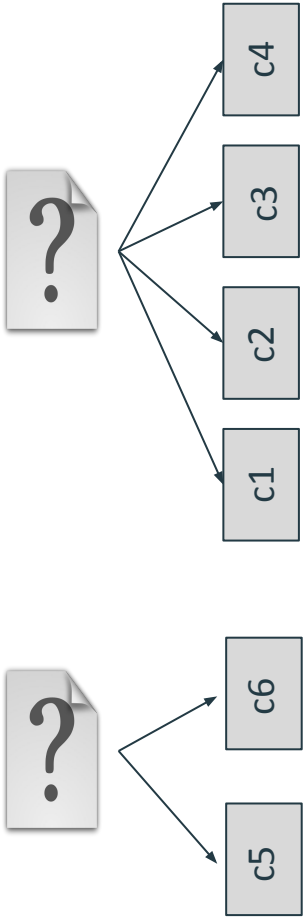
Text Processing & Integers Sorting

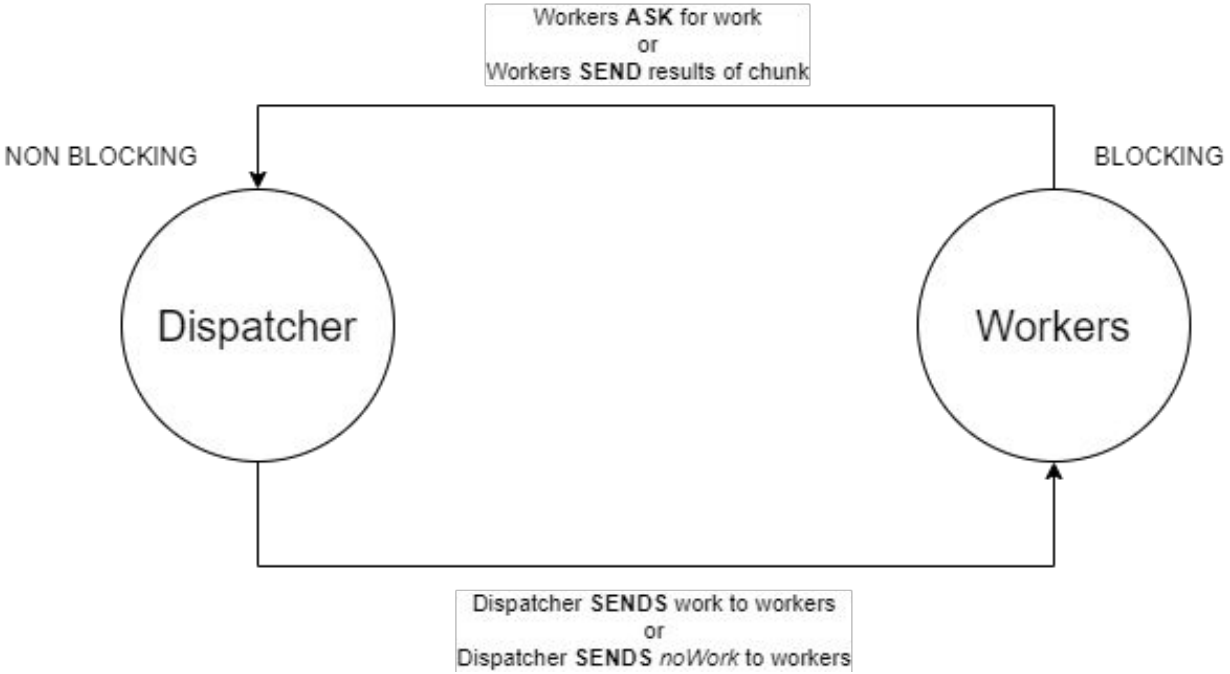
CLE2_T3G1 - Guilherme Antunes (103600) - Pedro Rasinhas (103541)

Prof. António Rui de Oliveira e Silva Borges

Text Processing



Dispatcher is Non Blocking;
Worker processes are Blocking;

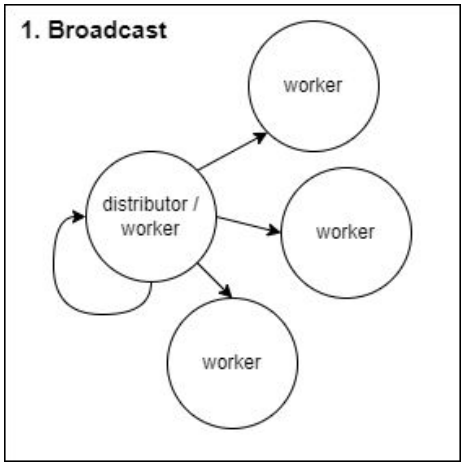


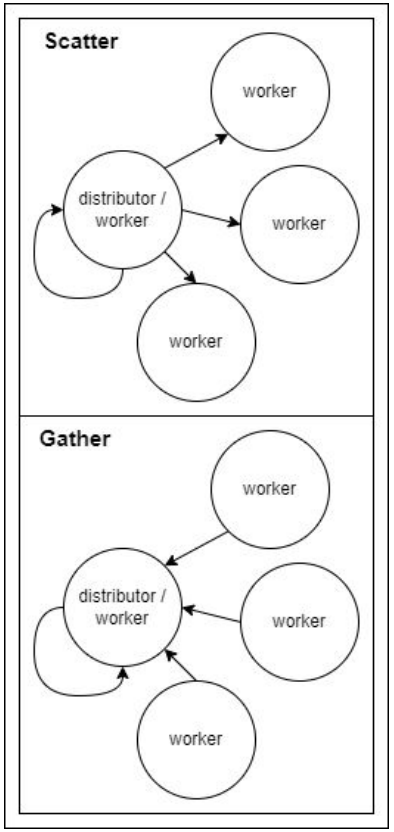
Text Processing - Results (all files)

Buffer/Chunk Size (Approximated values)	N of Processes	Avg Time Elapsed (s)*
4kb	3	0.142
	5	0.143
	9	0.109
8kb	3	0.121
	5	0.132
	9	0.130

*Average of 10 runs;

Integers Sorting





Implementation:

- Distributor broadcasts to the workers the sorting direction and the size of the array
- Scatter the array to all active workers.
- Each worker sorts its subarray for the first time; in subsequent iterations, it merges two subarrays.
- Gather the array from all active workers.
- Reduce the number of active processes by half.
- Repeat the process until only one process remains and the initial array is sorted

Integers Sorting - Results

N of Processes	1	2	4	8
dataSeq256K.bin				
Avg Time (s)*	0.091	0.058	0.034	0.053
dataSeq1M.bin				
Avg Time (s)*	0.394	0.241	0.186	0.200
dataSeq16M.bin				
Avg Time (s)*	8.007	4.973	3.606	3.739

*Average of 10 runs;

Integers Sorting - Conclusions

- Between 1 and 4 processes, we observed that the execution time decreases with the increase in the number of processes.

- In all cases, increasing from 4 to 8 processes does not result in an improvement in execution time. This is due to the use of blocking operations such as Broadcast, Scatter, and Gather, which require the distributor to wait for all workers to send a subarray or acknowledge receipt of messages. Each worker can only send a subarray after processing it properly. Increasing the number of processes entails waiting for more workers to complete their tasks, leading to increased waiting time and decreased program efficiency.