deti universidade de aveiro
departamento de eletrónica,
telecomunicações e informática

Assignment 2

- João Leite, nº 115041
- Luís Batista, nº 115279

Problem 1

Entities diagram

Workflow

Initialize some variables and MPI (using **MPI_Init**, **MPI_Comm_rank**, **MPI_Comm_size**);

1. Dispatcher's tasks:

Proccess the command line;

Distribute chunks of data to the workers (using **MPI_Isend**);

Get ready to receive responses (using **MPI_Irecv**);

Test if anyone completed his job and, if it's true, save the results (using **MPI_Test**);

Get remaining results and send a message for workers to terminate (using **MPI_Send**);

2. Workers' tasks:

Wait for work (using **MPI_Recv**);

Proccess chunk;

Send results (using **MPI_Send**);

Terminate proccess (using **MPI_Finalize**);

Problem 1

Results

PC1 Specs:

Lubuntu VM
4 cores

1 worker		2 workers		4 workers		8 workers	
Mean	Standard deviation	Mean	Standard deviation	Mean	Standard deviation	Mean	Standard deviation
15.933 ms	2.358 ms	17.545 ms	4.722 ms	11.612 ms	1.059 ms	869.795 ms	410.996 ms

PC2 Specs:

Ubuntu VM
2 cores

1 worker		2 workers		4 workers		8 workers	
Mean	Standard deviation	Mean	Standard deviation	Mean	Standard deviation	Mean	Standard deviation
12.973 ms	6.596 ms	7.613 ms	0.832 ms	39.461 ms	23.182 ms	40.792 ms	16.883 ms

*(**NOTE**: both mean and standard deviation results were calculated by running the program 5 times)*

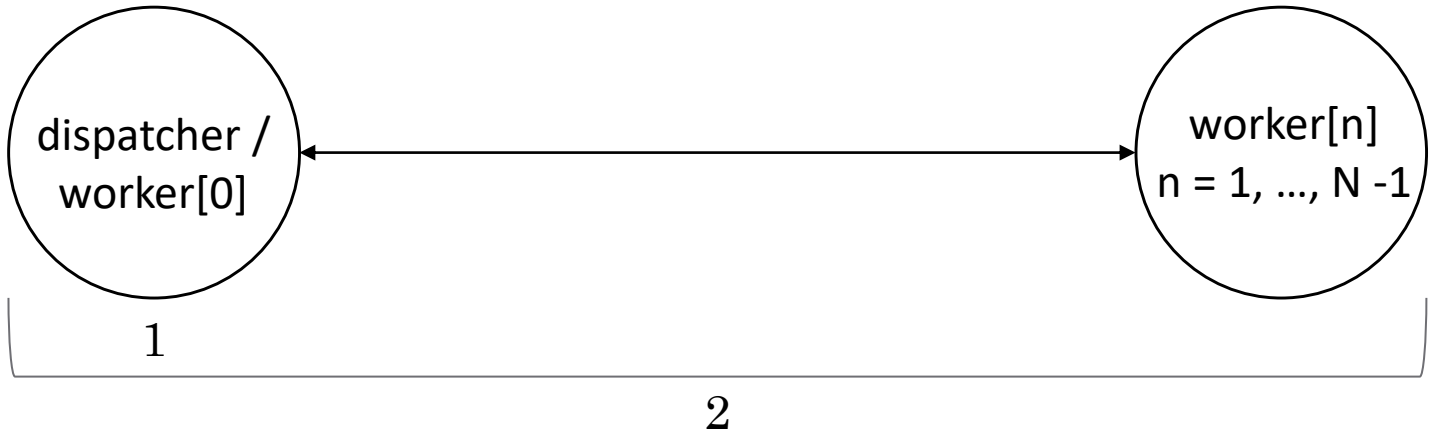
Problem 1

Conclusions

- **Expected results**
  - Have improvenements until reaching the maximum cores of each PC;
  - Get the best performance in the maximum size of cores that each PC supports;
  - With more than the actual number of available CPUs, it would be expected for the program to get longer times;
- **Outcome**
  - The best times were obtained with the number of cores that we were expecting:
    - With 4 workers for the PC1;
    - With 2 workers for the PC2;
  - Some operating system conditionings may have affected the results (in this cases, the standard deviation indicate us those “anomalies”).
    - For example: In PC1, the result for the 2 cores was worst than using a single core, but the standard deviation was the double.

Problem 2

Entities diagram



Workflow

1. Dispatcher's tasks:

Get sequence length and send it to all workers (using **MPI_Broadcast**);
Scatter subsequences of integers to all workers (using **MPI_Scatter**);
Validate sequence;

2. Workers' tasks:

Initialize some variables and MPI (using **MPI_Init**, **MPI_Comm_rank**, **MPI_Comm_size**);
Create and handle the communication (using **MPI_Comm_group**, **MPI_Group_incl**, **MPI_Comm_create**);
Sort subsequence;
Gather the ordered subsequence (using **MPI_Gather**);
Go to the next level until getting the final sequence;

Terminate proccess when they are not needed anymore (using **MPI_Finalize**);

Problem 2

Results

PC1 Specs:

Lubuntu VM
4 cores

Binary file	1 worker		2 workers		4 workers		8 workers	
	Mean	Standard deviation	Mean	Standard deviation	Mean	Standard deviation	Mean	Standard deviation
datSeq32.bin	0.021 ms	0.003 ms	0.039 ms	0.006 ms	0.053 ms	0.032 ms	73.449 ms	146.705 ms
datSeq256K.bin	61.542 ms	1.248 ms	38.090 ms	6.166 ms	30.856 ms	7.515 ms	59.804 ms	79.884 ms
datSeq1M.bin	298.796 ms	17.305 ms	178.738 ms	23.191 ms	219.899 ms	107.156 ms	100.495 ms	8.299 ms
datSeq16M.bin	5.875 s	0.132 s	3.383 s	0.192 s	2.335 s	0.101 s	2.654 s	0.070 s

PC2 Specs:

Ubuntu VM
2 cores

Binary file	1 worker		2 workers		4 workers		8 workers	
	Mean	Standard deviation	Mean	Standard deviation	Mean	Standard deviation	Mean	Standard deviation
datSeq32.bin	0.084 ms	0.004 ms	0.109 ms	0.013 ms	0.235 ms	0.107 ms	39.888 ms	10.400 ms
datSeq256K.bin	94.068 ms	11.083 ms	64.103 ms	7.478 ms	76.072 ms	27.637 ms	7.683 s	0.519 s
datSeq1M.bin	406.715 ms	21.009 ms	242.246 ms	1.274 ms	212.104 ms	4.918 ms	32.059 s	1.930 s
datSeq16M.bin	8.152 s	0.182 s	5.445 s	0.139 s	5.068 s	0.296 s	-	-

*(**NOTE**: both mean and standard deviation results were calculated by running the program 5 times)*

Problem 2

Conclusions

- Expected results:
  - Have improvenements until reaching the maximum cores of each PC;
  - Get the best performance in the maximum size of cores that each PC supports;
  - With more than the actual number of available CPUs, it would be expected for the program to get longer times;
- Outcome:
  - For the dataSeq32.bin file the results are not very meaningful since we are dealing with a very short sequence and the communication group operations time ends up affecting the performance.
  - Overall, all the remaining results follow the expectations whereas, for the outsiders, the standard deviation can explain any uncommon value.