Parallel and Distributed Programing Assignment 2

Fredrik Gustafsson

April 27, 2020

1. Implementation

To parallelise the matrix-matrix multiplication a solution based upon row-wise partitioning was chosen. Since we assume that the number of rows are to be divisible by the number of PEs the load will be well balanced.When reading the two matrices A is read row-wise while B is read column wise. The partitioned data is then broadcasted from the master PE, defined as PE with rank 0, to the other PEs. Each PE then calculates its local element of the resulting C matrix, which is done by sending and receiving parts of the B matrix. Finally the master gathers all local results into a global matrix, finally the output is written to a file by the master.

2. Performance Meassurements

To meassure the performance of the program a start time meassurement was taken after the reading of the file and an end meassurement was taken after the gatering into a global matrix. The code was then run on the rackham cluster five times for each configuration to get an average time.

2.1. Strong Scaling

To meassure the strong scaling, the speedup from increasing the number of PEs, one, two, four, eight and 16 PEs were tested using the file with containing a matrix of 7488 rows and columns. The resulting average time of fime executions for each number of PEs is shown in Table 1

Number of PEs	Average Time
1	520.0240446
2	306.2658
4	166.1116
8	102.2257
16	76.0155

Table 1: Average time for different number of PEs

A plot of the achieved speedup together with the teorethical speedup is shown in Figure 1 below.

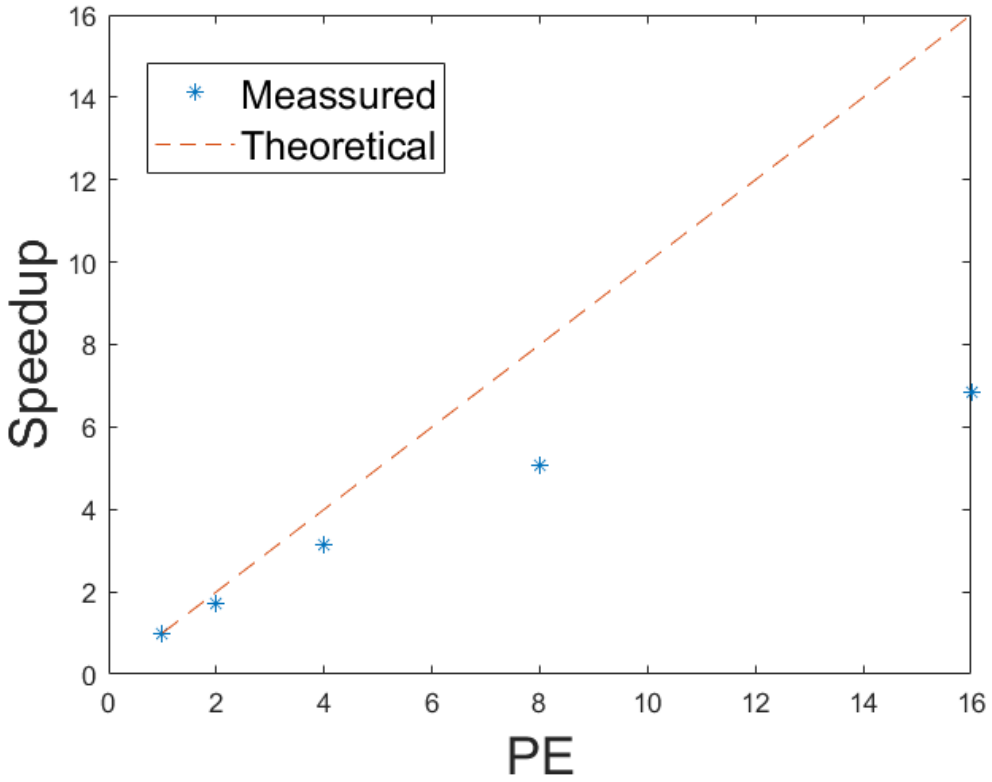

Figure 1: Speedup when using one, two, four, eight and 16 PE.

2.2. Weak Scaling

To investigate the weak scaling we once again investigate how different number of PEs affect the computation but this time when varying the problem size. The omplexity of multiplying two matrices is $O(n^3)$ hence the amount of PEs needed, compared to using a single PE, can be calculated as $\frac{n_i^3}{n_1^3}$, under the assumption of ideal theoretical weak scaling. The average time from five executions are shown in Table 2 below.

PEs	n	$\frac{n_i^3}{n_1^3}$	Average Time
1	3600	1	50.98224
4	5716	4.003	69.8179
9	7488	8.999	92.
16	9072	16.003	121.8174

Table 2: Weak scaling of time for different amount of PEs and matrice sizes.

3. Discussion

The parallelisation is not perfect as seen in figure 1, or the two tables. This might be due to communication overhead decraesing the parallel performance since the communication is done in a circular fasion. This then results in diminishing improvements as we increase the number of PEs appearing to converge towards a speedup of seven.f Furthermore the timing includes the broadcasting and scattering of data as well as the final gathering by the master PE these could also potentially affect the parallel performance however this should be fairly minor. One improvement might be a 2D partitioning of the data instead of the one dimensional chosen in this assignment.