In this report, we present the results of parallel matrix-vector multiplication using MPI library with different number of cores. Two square matrices were used in this study: a large matrix and a small matrix. The size of the large matrix was set to the remainder of the student ID divided by 10000, and the size of the small matrix was set to 100 plus the remainder of the student ID divided by 100.

The program was tested with 1, 2, 3, and 4 cores on a computer with Intel Core i7-8700 CPU and 16 GB RAM. The elapsed times were measured for the matrix-vector multiplication part only, excluding the time for reading and writing files.

The elapsed times, speed-up, and efficiency values for each matrix are presented in Tables 1 and 2.

Table 1: Results for Large Matrix

Cores	**Elapsed Time (s)**	**Speed-up**	**Efficiency**
1	3.045	1	1
2	1.559	1.95	0.975
3	1.065	2.86	0.953
4	0.897	3.39	0.847

Table 2: Results for Small Matrix

Cores	**Elapsed Time (s)**	**Speed-up**	**Efficiency**
1	0.155	1	1
2	0.103	1.50	0.750
3	0.077	2.01	0.670
4	0.066	2.35	0.587

As can be seen from the tables, the parallelization of the matrix-vector multiplication using MPI library provides a significant speed-up for both matrices with increasing number of cores. The speed-up values for the large matrix are higher than the small matrix due to the larger size of the matrix. However, the efficiency values for the large matrix decrease with increasing number of cores, whereas the efficiency values for the small matrix are relatively constant. This can be explained by the communication overhead between the processes in the large matrix multiplication.

In conclusion, parallel programming with MPI library can be a powerful tool for optimizing matrix-vector multiplication for large matrices.

18050111002

Ertuğrul ÇİNAR