COMP 529 **Shukhrat Khuseynov (0070495)** **08.06.2020**

Assignment 3 (Report)

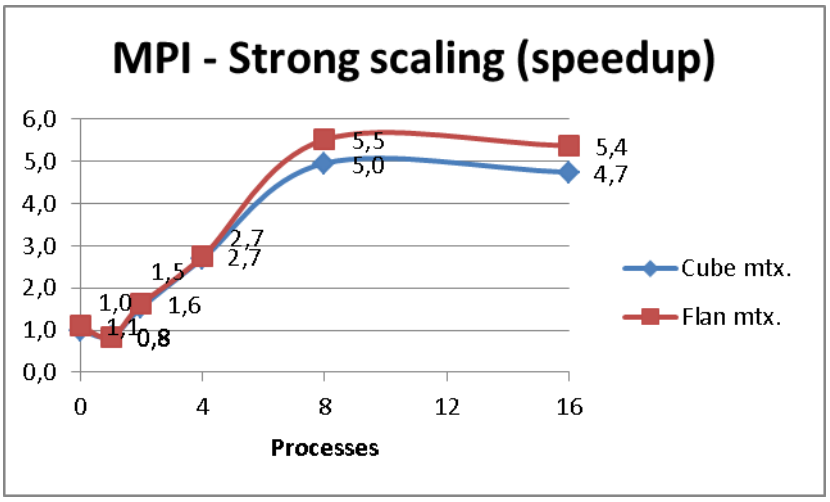
The MPI implementation was developed for the iterative sparse matrix-vector multiplication problem using compressed storage row format. Two required parts specified in the assignment description were completed and saved with relevant names: main_p1.cpp and main_p2.cpp. The attempt to do the optional third part was unsuccessful requiring more time and, therefore, it is not included. The performance study was done in a single job on rk01 node. The job script and its output are saved as submit_job.sh and spmv.out, respectively.

The first, main, part consisted of establishing multiprocessing mainly involving in communication between the processes and the root process, which was different from others in the sense that it was dealing with reading matrix, measuring time, processing command line arguments, possessing and distributing the data via collective functions, such as MPI_Bcast, MPI_Scatterv, and MPI_Allgatherv. MPI_Barrier was only used to measure the time more accurately. Some of the important variables, such as N, M and rowptr array, were broadcasted among the processes at the beginning. One critical task was to distribute the number of original matrix rows between the processes, which is not guaranteed to be divisible. This also required unequal distribution for scattering and gathering, therefore MPI_Scatterv and MPI_Allgatherv were utilized instead of MPI_Scatter and MPI_Allgather functions. The number of rows and the start point for each process were computed and saved as arrays to be used in the following computations and as an input for collective functions. Using this information together with rowptr, the val and colidx arrays were scattered between the processes. In each time step of the main computation, after calculating partial results of rhs, it was gathered and broadcasted between all the processes in a single function.

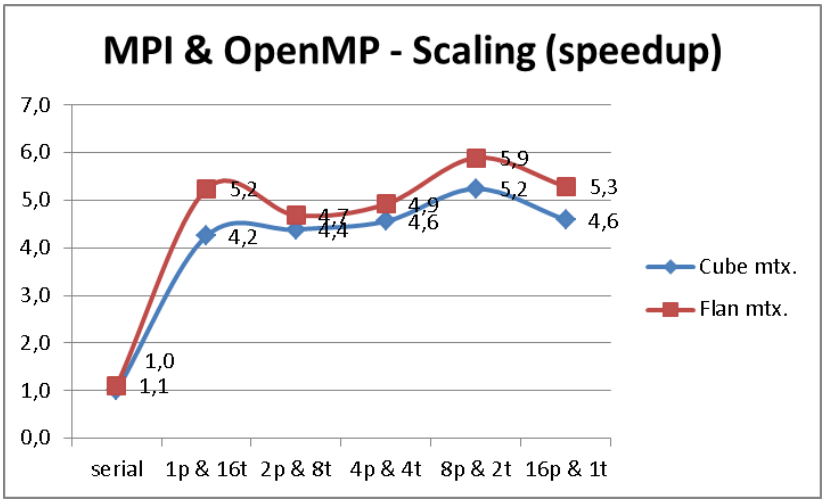
In the second part, multithreaded parallelism of OpenMP was embedded to the multiprocessing given in MPI code of the first part. Due to high dependency in the sequence of the code, only data parallelism, rather than task parallelism, was possible to implement. The point of concentration here was the main computation section of matrix-vector multiplication by each process, which comprises two nested loops. The attempt to use omp parallel for collapse(2) feature with critical section to sum up the results was not efficient and was very slow due to high overhead. Hence, only single parallel for loop was used there. Parallel for was also employed for few other loops with high N, which is less effective. In this part the number of threads is also given as a command line argument, standing after the name of the executable file, i. e. "`mpirun -np 2 build/spmv_p2 8 Flan_1565/Flan_1565.mtx 20`" (8 threads).

The correctness of these parts was ensured by printing some elements of rhs matrix, running with small_test1.mtx, small_test2.mtx, Flan_1565.mtx, and Cube_Coup_dt6.mtx and being compared to the results of their serial implementation. All of them were accurate and identical. As a bottleneck, for all the codes, the slowest portion is reading and processing the

input data, taking more than a minute for the given big matrices, although it is not considered in the time count. The results of performance studies are given below:



It can be seen that in the first experiment, the highest speedup is achieved by 8 processes. 16 processes are slower possibly due to some parallelization overhead or machine limitations. Apparently, one process MPI program is slower than its serial equivalent.



The second experiment concludes with the highest speedup of 8 processes with 2 threads. Moreover, there is one another peak, which is a bit lower; it is a combination of 1 process with 16 threads.

In conclusion, it was a very interesting assignment for learning and applying the MPI parallel implementations and its hybrid combination with OpenMP. I learned a lot from this course and the assignments in general. The topics were fascinating.