Parallel Programming Project

Data Parallelization of the Edit Distance Algorithm
-Third method-
Using multiple communicators

عبد الله احمد عبدالباسط محمد
Sec 1 CS

Dr. Mohmmad Yousef Bassiouni

Introduction

This is a dynamic program that works with any number of communicators, processes, threads, database and queries. The length of the datababse, queries must not exceed the maxDatabaseLength, maxQueryLength respectively.

I attached a VS2022 solution, image for the code ,pdf for the code to make it easy to read the code, I also wrote a comment for every line in the code, Please open the “The code.pdf” file to read the code from.

The code

It’s a c program that contains five functions, the main and four other functions.

Line 1 – 20 includes the libraries I am using, declares some constants, declares the arrays that holds the database and the queries.

Line 23-31 I created a struct that will hold a result value and the index of the string in the database, some functions definitions that declared at the end of the program.

Line 37-49 declaring some variables and pointers to be used in the program.

Line 52-54 starting with the MPI, getting the comm_sz, myrank for every process.

Line 57-66 process 0 in world comm gets the number of communicators that the processes will be divided over them, the number of threads the process will fork.

Line 69-70 broad casting the commCount, numberOfThreads to all the processes.

Line 73 calculating the number of processes every communicator will take.

Line 75-85 checking that the number of processes is greater than or equal to the number of communicators entered.

Line 86-92 handling the case that the processes is not divisible by the communicators.

Line 94-100 splitting the world comm into multiple communicators, getting the new rank,new size.

Line 103-148 process 0 from the world reads the database and the queries from the files using the readFromFile function, handling the issue that the database is not divisible by the number of processes in the communicator, the issue that the queries is not divisible by the number of communicators.

Line 151-152 broad casting the length of the database and the length of the queries.
Line 154-166 claculating the amount of database,queries elements, the starting and ending of every process. i.e dividing the data.

Line 170 – 191 process 0 of the first comm sends the database and queries part to process 0 of every new comm.

Line 198-195 process 0 of every comm scatters the database over the processes in its comm, broad casts the queries to all the processes in its comm.

Line 201-209 all the processes allocates a memory to store the results in.

Line 208-223 every process forks a number of threads using OMP, working in its part of the queries over its part of the database using editDist function.

Line 226-230 every process sorts its results using compare function.

Line 233-241 process 0 of every comm allocates a memory space to gathers the results from all the processes in its comm.

Line 244-248 process 0 of every comm gathers the results from all the processes in its comm.

Line 261-283 process 0 of the first comm receives the results from process 0 of every comm that it gathered from all the processes.

Line 287-301 printing the results.