Multi-Threaded Matrix Multiplication with Thread affinity

Structure

This directory contains 3 sub-dirs:

- `Input Files` : contains input file `input.txt`
- `Output Files` : contains 2 output files `output1.txt` and `output2.txt`
- `Source Files` : contains 2 source files `Assgn2_Mixed_Src-CO22BTECH11006.cpp` and `Assgn2_Chunk_Src-CO22BTECH11006.cpp`

It also contains 2 additional files

- `Assgn2_Readme-CO22BTECH11006.md` : contains overview of the program
- `Assgn2_Report-CO22BTECH11006.pdf` : contains complete report of the assignment with experiments.

Overview

The program allows you to specify the size of the matrix (`n`), the number of threads (`k`), the number of logical cores (`c`) and the number of bounded threads (`bt`). It reads the matrix from an input file, performs square of matrix using the specified methods, and records the total execution time, the average execution time of bounded threads and the average execution time of unbounded threads.

Prerequisites

- C++ compiler
- pthread library (for multi-threading)
- Input file (`input.txt`) containing matrix data
- Output file (`output1.txt`) to record results
- Output file (`output2.txt`) to record results

Compilation and Execution

To compile the program, navigate to the program's directory and use the following command:

```
g++ -O3 Assgn2_Chunk_Src-CO22BTECH11006.cpp -lpthread -o matrix_multiplication
```

Here, O3 is used to optimize the CPU performance. All results recorded in report are obtained by using O3.

To run the program, execute the compiled binary:

```
./matrix_multiplication
```

Input

Create the `input.txt` file in the `Input Files` directory. The file should be in following format:

```
 N K C BT
A[0][0] A[0][1] ... A[0][n-1]
A[1][0] A[1][1] ... A[1][n-1]
...
A[n-1][0] A[n-1][1] ... A[n-1][n-1]
```

here, N : size of the matrix K : Total number of threads C : number of logical cores (Which is 12 in my system) BT : number of bounded threads

Output

The program will output the results in the `output1.txt` or `output2.txt` file in the `Output Files` directory depending on which method is used. For Chunk method, the output file will be `output1.txt` . For Mixed method, the output file will be `output2.txt` .

In the output file, the first line will contain the size of the matrix, the second line will contain the number of threads used. The third line will contain the number of bounded threads. The fourth line will contain the number of logical cores used. The fifth line will contain the number of cores which assigned to bounded threads.

After that, the program will output total execution time. Followed by the average execution time of bounded threads and average execution time of unbounded threads.

The program then displays which core is assigned to which bounded thread.

The program then prints the square matrix (A*A).

At the end, the program will print the execution time for each thread.

Notes

- Ensure that the `Input Files` and `Output Files` directories exist in the current directory or modify the paths accordingly.
- The performance and efficiency can vary based on the system's hardware and available resources.

Author

- Name: Om Dave
- Roll No.: CO22BTECH11006