

Programming Assignment 1: Measuring Matrix Sparsity

Course: CS5300 - Parallel & Concurrent Programming

Assignment: Measuring Matrix Sparsity using Parallel Programming Techniques

Author: Om Dave (CO22BTECH11006)

Submission Date: 14th August 2024

Overview

This assignment implements three parallel methods for calculating the sparsity of a square matrix using C++ and POSIX Threads (pthread). The methods implemented are:

1. **Chunk Method:** The matrix is divided into equal-sized chunks, and each thread processes a specific chunk.
2. **Mixed Method:** Threads are assigned rows in a round-robin fashion across the entire matrix.
3. **Dynamic Method:** Threads dynamically allocate rows to process as they become available.

Additionally, a **Mixed-Chunk** method, a hybrid of the Chunk and Mixed methods, has also been implemented.

Files and Directory Structure

- **Assgn1-CO22BTECH11006/:** Contains the source code for each method.
 - **Assgn1-Chunk-CO22BTECH11006.cpp:** Source code for the Chunk method.
 - **Assgn1-Mixed-CO22BTECH11006.cpp:** Source code for the Mixed method.
 - **Assgn1-Dynamic-CO22BTECH11006.cpp:** Source code for the Dynamic method.
 - **Assgn1-Mixed-Chunk-CO22BTECH11006.cpp:** Source code for the Mixed-Chunk method.
- **input.txt:** Input file containing matrix dimensions and data.
- **output.txt:** Output file where the results of the program execution are stored.

How to Compile and Run

Compilation

To compile the source code for each method, navigate to the **Assgn1-CO22BTECH11006/** directory and use the following command (For eg for chunk method):

```
g++ -o chunk Assgn1-Chunk-CO22BTECH11006.cpp -lpthread
```

Execution

To execute the compiled programs, run the following commands:

```
./chunk
```

Input File

Format The input file (**input.txt**) should contain the following parameters:

```
N S K rowInc
Matrix Data
```

Where:

- **N** is the size of the matrix (NxN).
- **S** is the sparsity percentage.
- **K** is the number of threads.
- **rowInc** is the row increment (relevant for the Dynamic and Mixed Chunk method).

The **Matrix Data** section should contain **N** rows, each with **N** integer values representing the matrix elements.

Example Input

```
5000 40 16 50
0 1 0 ... 1 0
1 0 0 ... 1 0
...
0 0 1 ... 0 1
```

Output

Each program generates an output file (**output.txt**) containing the following information:

- Time taken to compute the sparsity.

- Total number of zero-valued elements.
- Number of zero-valued elements counted by each thread.

Example Output

```
Time taken to count the number of zeros: 123 ms
Total Number of zero-valued elements in the matrix: 5920000
Number of zero-valued elements counted by thread1: 370000
Number of zero-valued elements counted by thread2: 370000
...
Number of zero-valued elements counted by thread16: 370000
```