ECEN 5033 Concurrent Programming

LAB1 Report

Parallelization Strategies:

1. Merge Sort
    - The given array is divided into sub arrays equal to the given number of threads. Size of the subarray is total elements divided by number of threads.
    - Each thread then sorts and merges the assigned sub array.
    - As soon as all threads joins, all the sorted subarrays are merged to main array.
2. Bucket Sort
    - Again, the given array is divided into sub arrays equal to the given number of threads. Size of the subarray is total elements divided by number of threads.
    - Each thread thread then works on putting the elements into the buckets based on the derived hash value.
    - Hash value is calculated using the equation: (number of buckets * array element)/max element of the array
    - Number of buckets is equal to the number of threads. STL multisets are used as a bucket.
    - As soon as all threads joins, all the buckets are merged into main array.

Code Organization

- Handling Command-line arguments:
  getopt_long() function is used to handle required command-line arguments, --name, -t for number of threads, -o for output file and --alg for selecting the algorithm.
- As the input size (number of integers in the file) is unknown realloc is used to dynamically allocate memory. And thus handling various input sizes also becomes easier.
- After reading the input file and allocating memory for each integer, the array will be divided based on the given number of threads and will be sorted using selected sorting algorithm.
- Followed by printing sorted data to the output file, if given by the command line else to the stdout.

Description of files submitted

- Makefile: as the name suggests it is a Makefile to compile and build the application. Use **make** on the terminal to do the same.
- Mysort.cpp: this is the source file for the lab. It has the main() function and all other required functions to fulfill the lab requirements.
- Random.sh: this file is a bash script used for generating a test file with random number of integers.
- Log.h: a simple logging system for different log levels. Disabled by default to save time from used by printfs.

Compilation Instructions

- After downloading the submitted .zip file, extract and go into the directory.
- Use the **make** command to build and create an executable named **mysort.**

Execution Instructions

- **./mysort --name** prints student name
- **./mysort inputfile.txt -t [num of threads] -o output.txt --alg [merge or bucket]**
  Case 1: to test Merge Sort
  ./mysort inputfile.txt -t 50 -o output.txt --alg merge
  Case 2: to test Bucket Sort
  ./mysort inputfile.txt -t 50 -o output.txt --alg bucket