AMATH 483 / 583 - HW4

May 6, 2023

1 Coding, coding, coding

1. (+15) IO Bandwidth. On a computer of your choice, write a C++ function that writes type double binary square matrices in column major order to file. Increase the dimension of matrices written as 32, 64, 128, ... 16384 (2GB), and measure the time required to complete each write. Similarly, write a C++ function that reads the type double matrix data from file into memory, and measure the time required to complete each read for these dimensions. Make a single plot of your *read* and *write* measurements with the bandwidth (bytes per second) on the y-axis, and the problem dimension on the x-axis. Submit your functions with test code and plot.

2. (+20) Length of a graph of $f(x)$ on $[a, b]$ is $L = \int_a^b \sqrt{1 + (f'(x))^2}dx$. Given $f(x) = ln(x) - \frac{1}{8}x^2$ on $[1, 6]$, find the length of f on the interval analytically. You will need this to calculate the error in your code. Write a C++ code parallelized using the C++ threads library that numerically evaluates the length of this function on the interval. Your code should accept the number of points used to partition the interval and the number of threads to spawn. Plot the strong scaling efficiency (time versus thread count) of your code for $n = 1.e8$ partition points for 1 to 6 threads (or more). Plot the numerical error for $n = 10$ to $n = 1.e6$ partition points, increasing by factors of 10 each time. Submit your code and plots.

3. (+20) Length of a graph of $f(x)$ on $[a, b]$ is $L = \int_a^b \sqrt{1 + (f'(x))^2}dx$. Given $f(x) = ln(x) - \frac{1}{8}x^2$ on $[1, 6]$, find the length of f on the interval analytically. Write a C++ code parallelized using MPI that numerically evaluates the length of this function on the interval. Your code should accept the number of points used to partition the interval. Plot the strong scaling efficiency (time versus process count) of your code for $n = 1.e8$ partition points for 1 to 40 processes using HYAK. Plot the numerical error for $n = 10$ to $n = 1.e6$ partition points, increasing by factors of 10 each time. Submit your code and plots.

4. (+20) Communication Bandwidth. Conduct this problem on HYAK. Write a MPI C++ template function using point-to-point communication that implements broadcast. The api I want you to use is provided. Compare the performance of your function to *int MPI_Bcast(void *buffer, int count, MPI_Datatype datatype, int root, MPI_Comm comm)* for messages of size 8 bytes to 128MB increasing in multiples of size 2 on 40 processes. Make a plot of your measurements with the bandwidth (bytes per second) on the y-axis, and the message size in bytes on the x-axis.

   -
     ```
     template <typename T>
     void my_broadcast(T* data, int count, int root, MPI_Comm comm);
     ```

5. (+25) Implement a threaded elevator scheduler for a 20-story apartment building with 3 elevators. The building has 10 apartments on each floor. The program should use the C++ threads library to simulate the behavior of the elevators and residents of the building. Your code should take as input the number of people to service. Once servicing all the people completes, your code should exit cleanly. Please print events to *cout* in a manner that is thread safe.

   - randomly select the current floor and destination floor for each person
   - each elevator will start from the ground floor
   - each person will issue a thread in the *detach* state (i.e. *thread(person, i).detach();*)
   - each elevator will be it's own thread
   - keep track of the current floor and direction of each elevator
   - keep track of the destinations of the residents who are waiting for elevators on each floor
   - determine which elevator should respond to a given request, based on its current location, direction, and the number of passengers it is already carrying

- move the elevators up or down to their next destination, picking up and dropping off passengers as necessary
- handle multiple residents and elevator requests at the same time, using threads to manage the concurrent operations (I used mutex only for this)
- print events to screen as they occur - see below for example output from my version; in the example 100 people were enetered and elevator 0 serviced 41 people, elevator 1 serviced 31 people, and elevator 2 serviced 28 people

```
Person 0 wants to go from floor 7 to floor 9
Person 0 entered elevator 0
Person 1 wants to go from floor 13 to floor 18
Person 1 entered elevator 0
Elevator 0 moving from floor 1 to floor 7
Elevator 0 moving from floor 2 to floor 9
Elevator 0 moving from floor 3 to floor 13
Elevator 0 moving from floor 4 to floor 18
Person 2 wants to go from floor 10 to floor 12
Person 2 entered elevator 0
...
...
Elevator 2 moving from floor 11 to floor 2
Elevator 2 moving from floor 10 to floor 2
Person 96 wants to go from floor 11 to floor 19
Person 96 entered elevator 2
Elevator 2 moving from floor 11 to floor 11
Elevator 2 moving from floor 12 to floor 19
Person 97 wants to go from floor 12 to floor 4
Person 97 entered elevator 2
Person 98 wants to go from floor 15 to floor 18
Person 98 entered elevator 2
Elevator 2 moving from floor 11 to floor 4
Elevator 2 moving from floor 12 to floor 15
Person 99 wants to go from floor 16 to floor 12
Person 99 entered elevator 2
Job completed!
Elevator 0 has finished servicing all people.
Elevator 2 moving from floor 13 to floor 18
Elevator 1 has finished servicing all people.
Elevator 2 moving from floor 14 to floor 16
Elevator 2 moving from floor 13 to floor 12
Elevator 2 has finished servicing all people.
```