ECE574: Cluster Computing – Homework 6
MPI

Due: Friday, 10 March 2023, 5:00pm

1. **Background**

   - In this homework we will take the sobel code from earlier homeworks and parallelize it using MPI.

2. **Setup**

   - For this assignment, log into the same Haswell-EP machine we used in previous homeworks. As a reminder, use the username handed out in class and ssh in like this
     `ssh -p 2131 username@weaver-lab.eece.maine.edu`
   - Download the code template from the webpage. You can do this directly via
     `wget http://web.eece.maine.edu/~vweaver/classes/ece574/ece574_hw6_code.tar.gz`
     to avoid the hassle of copying it back and forth.
   - Decompress the code
     `tar -xzvf ece574_hw6_code.tar.gz`
   - Run `make` to compile the code.
   - You may use your own code from a previous assignment as a basis for this assignment. (Alternately some really poorly-optimized sample code is provided).

3. **Coarse-grained Code (8 points)**

   Use MPI to parallelize your code. Use the sample code, or you might want to use one of your previous assignments as a basis.

   If not using the sample code you will need to make sure your code includes `mpi.h` and that it calls `MPI_Init()` at the beginning and `MPI_Finalize()` at the end.

   Edit the file `sobel_coarse.c`

   Be sure to comment your code!

4. **A suggested first (coarse) implementation**

   (a) Initialization (note the provided code already does these for you)

   - Be sure `MPI_Init()` is called at the start
   - Be sure `MPI_Finalize()` is called at the end
   - Use `MPI_Comm_size()` to get the total number of ranks
   - Use `MPI_Comm_rank()` to get the rank number of the currently running code
   - In rank 0 print a debug message saying how many ranks there are.
   - Using `MPI_Wtime()` record the times for load/convolve/combine/store and print these at the end in rank 0.

(b) Load and Broadcast the Image Data

- Be sure to only load the jpeg in rank 0
- Create an integer array with three integers. Set these to the values of `image.xsize`, `image.ysize`, `image.depth`
- Use `MPI_Bcast()` to send this array from rank0 to all the other ranks
- Make sure the other ranks set their values of `image.xsize`, etc, from the array
- In non-rank 0 you will need to `calloc()` `image.pixels` (usually `load_jpeg()` does this but that doesn't get called on non-rank 0)
- Now use `MPI_Bcast()` to broadcast the `image.pixels` data from rank0 to all the other ranks. **Note:** You want to broadcast `image.pixels`, not the entire `image` struct as in MPI you can't send structs, just arrays).
- A `MPI_Bcast()` has an implicit barrier, so after this point all ranks should be in the same place, and all should have copies of the full image data.

(c) Do the Convolutions

- Modify `generic_convolve()` so it takes new `ystart` and `yend` parameters to iterate through.
- In the main function, set the ystart and yend values based on your rank number.
- **In each rank print the start/end numbers and verify that all y values are being calculated**
- Instead of convolving into `sobel_x`, convolve into a temporary result, perhaps use `new_image` to make the following gather step easier.

(d) Gather the results

- Use `MPI_Gather()` to get the results from all the ranks into rank0
- Note: you only want to gather the pixel data (the array of chars) not the structure containing it. So you want to gather `sobel_x.pixels` not `sobel_x`
- In the previous step we recommended you convolve into a temporary results rather directly into `sobel_x`. This is because MPI won't let a gather have the same source and destination (i.e. on rank0 you can't gather from all `sobel_x` into your own `sobel_x`
- Also note that `MPI_Gather()` will gather from the *start* of each buffer and put it in the proper place in the result. So you have to modify the convolve routine to store the output starting at offset 0, rather than at offset ystart. If you forget this, the bottom ((N-1)/N)th of your image will be blank.

(e) Run Combine

- On rank 0 alone, run combine.

(f) Output to File

- On rank 0 alone, output to file

5. **Handle tail end data (1 point)**

(a) Your code will likely only work if the image ysize is a multiple of the total rank number.

(b) Modify your code to handle this. Either by having rank0 alone calculate the tail values, or else by separately sending the data from one of the other ranks.

(c) This is listed separately as it's not super important for this homework, so worry about getting this working only if you are able to get all the rest going.

6. **Report your Results (1 point)**

   - Run on the Haswell-EP machine for 1, 2, 4, 8 and 16 threads and report the results, as well as reporting the speedup and parallel efficiency for the total time.
   - Run your code with:
     `sbatch -n X time_coarse.sh`
     where you replace X with the number of cores to use.
   - If (for fun) you want a bigger image to test with, try `/opt/ece574/jan_15_2017_high_res.jpg`

7. **Some Debugging Hints**

   - If you have puzzling results, debug at each step of the way.
   - Start by testing the N=1 case, then N=2 case
   - You can verify the initial `MPI_Bcast()` is working by adding code something like

     ```
     if (rank==1) {
                store_jpeg("out1.jpg",&image);
        }
     ```

     and verify that the output image is same as the input image.
   - One you verify the broadcast works, do the same for the separate sobelx and sobely results.
   - After verifying those looks good, do the same for after the gather. Make sure gather range is the same as your convolve ranges.
   - Other things to watch for:
     - If you get a diagonal pattern in the output, be sure you are gathering in even multiples of `xsize`
     - Be sure your limits are set properly. Print the limits out and verify they are being set properly.

8. **Submitting your work.**

   - Be sure to edit the README to include your name, as well as the timing results, and any notes you want to add about your something cool.
   - Run `make submit` and it should create a file called `hw06_submit.tar.gz`. E-mail this file to me.
   - e-mail the file to me by the homework deadline.