Sfwr Eng/Comp Sci 4F03 (Winter 2017)
Programming Assignment 4

1 Assignment

1.1 Problem

You are to design a program which will utilize MPI to apply an image processing effect (Blur Filter) to an image in parallel across multiple MPI processes.

1.2 Program Parameters

When "make" is called within the PA4 folder, a program pa4.x should be generated. The program should be called from the command line in the following manner:

mpirun -np n ./pa4.x r <inputFilename>.ppm <outputFilename>.ppm

- n The number of MPI processes to use.
- r The Blur radius in pixels.
- <inputFilename>.ppm is the name of the file to blur in .ppm image format.
- <outputFilename>.ppm is the name of the file to save the blurred image to in .ppm image format.

1.3 Specification

- The program has to utilize n MPI processes (n should be 2,3,4,8,16,32).
- Image dimensions do not need to divide evenly by number of MPI processes (i.e., 10 rows with 3 processes).
- The images will be in PPM format (easy to use image format perfect for C programming).
- You may use the supplied ppmFile.c/.h files to handle ppm files.
- The result image needs to be blurred in a square brush with radius r (see below).

- Every pixel needs to be blurred, you need to pay special attention to corner and edge pixels (i.e., pixel(0,0) would be blurred using a quarter brush, instead of the full brush since we do not have pixels before (0,0)).
- Your program should work with HD images (i.e., fox.ppm).

The following are additional specs to run:

- You need to ssh into either mills or moore, from there ssh into mpihost01 (The Head Node of our MPI cluster).
- To convert any image file to .ppm format, use the "convert" command on mills or moore (i.e., "convert image.bmp image.ppm").
- For more accurate timings, you should run tests using 4K images.
- Remember to use MPICC and MPIRUN to compile and run your programs, else they will run single threaded only.
- You may use the supplied .ppm files for testing and calibration of your program.

1.4 Blur Filter

The Blur Filter which you should implement will be a square brush average filter. Each pixel $p[x, y] = (r, g, b)$ has an x and y coordinate, and red r, blue b, and green g colour channels. r, g, b should be unsigned char with values ranging 0 - 255. During processing, for each channel, the average of the sum of p and its neighbours using bounds based on the Blur Radius r will be calculated.

The bounds are as following:
$minX = x - r$
$maxX = x + r$
$minY = y - r$
$maxY = y + r$

Please be aware that $minX, maxX, minY, maxY$ have to be within the boundaries of the image (i.e., an image with a width of 100 pixels cannot have a $maxX$ boundary of 102).

We can then calculate the new colour of p using:

$$p[x, y].c = \frac{\sum_{i=minX}^{i=maxX} \sum_{j=minY}^{j=maxY} p[i, j].c}{\sum \text{pixels added}}, c \in \{r, g, b\}$$

Please see Figure 1.

Figure 1: A pixel to be processed with some of its neighbours

1.5 Example

When the Blur filter is applied to an image, it will become blurry as shown in Figure 2. and Figure 3.

Figure 2: Original Image

Figure 3: Image after applying Blur Filter with Blue Radius = 10

1.6 Submission

Please include an Analysis Report comparing the performance of input image size vs process number, similar to slide 121 of the MPI Slides.
Include Speedup and Efficiency plots like slides 95/96 in the MPI Slides. Summarize and discuss your results.
Is your program Strongly Scaleable? Weakly Scaleable? Why or why not for both?
Save it as Report.pdf

Your solutions must be submitted by **11:59pm on Sunday April 9th, 2017** in the provided SVN folder (see below)

All your source files and makefile should be located in PA4.

Please ensure that the program will work on mpihost01.

1.7 Grading

- Up to 40% for blurring image single node
- Up to 60% for distributing to n nodes
- Up to 80% for blurring correctly on n nodes
- Up to 100% for correct Analysis Report

2 Setup SVN

You have been provided an SVN trunk located at:

```
https://websvn.cas.mcmaster.ca/4f03/<macid>

Where <macid> is your MacID used to log onto Mosaic and Avenue.
E.G. If your MacID is bob, the svn addess is:
https://websvn.cas.mcmaster.ca/4f03/bob

Check out a working directory of your SVN to your comptuer or
department machine (e.g. moore or mills).
Create Folders PA1, PA2, PA3, PA4, PA5, PA6. You structure should be:

<macid>
|->PA1
|->PA2
|->PA3
|->PA4
|->PA5
```

```
|->PA6

Use: svn checkout and svn commit
```