1)

a)

```
dgin@ecc-linux2:~/Documents/csci551/numeric-parallel-starter-code/hello_openmp$ ./hello_omp 4
Hello from OMP thread 11 of 16
Hello from OMP thread 0 of 16
Hello from OMP thread 5 of 16
Hello from OMP thread 6 of 16
Hello from OMP thread 7 of 16
Hello from OMP thread 8 of 16
Hello from OMP thread 9 of 16
Hello from OMP thread 12 of 16
Hello from OMP thread 13 of 16
Hello from OMP thread 14 of 16
Hello from OMP thread 15 of 16
Hello from OMP thread 10 of 16
Hello from OMP thread 4 of 16
Hello from OMP thread 1 of 16
Hello from OMP thread 2 of 16
Hello from OMP thread 3 of 16
dgin@ecc-linux2:~/Documents/csci551/numeric-parallel-starter-code/hello_openmp$
```

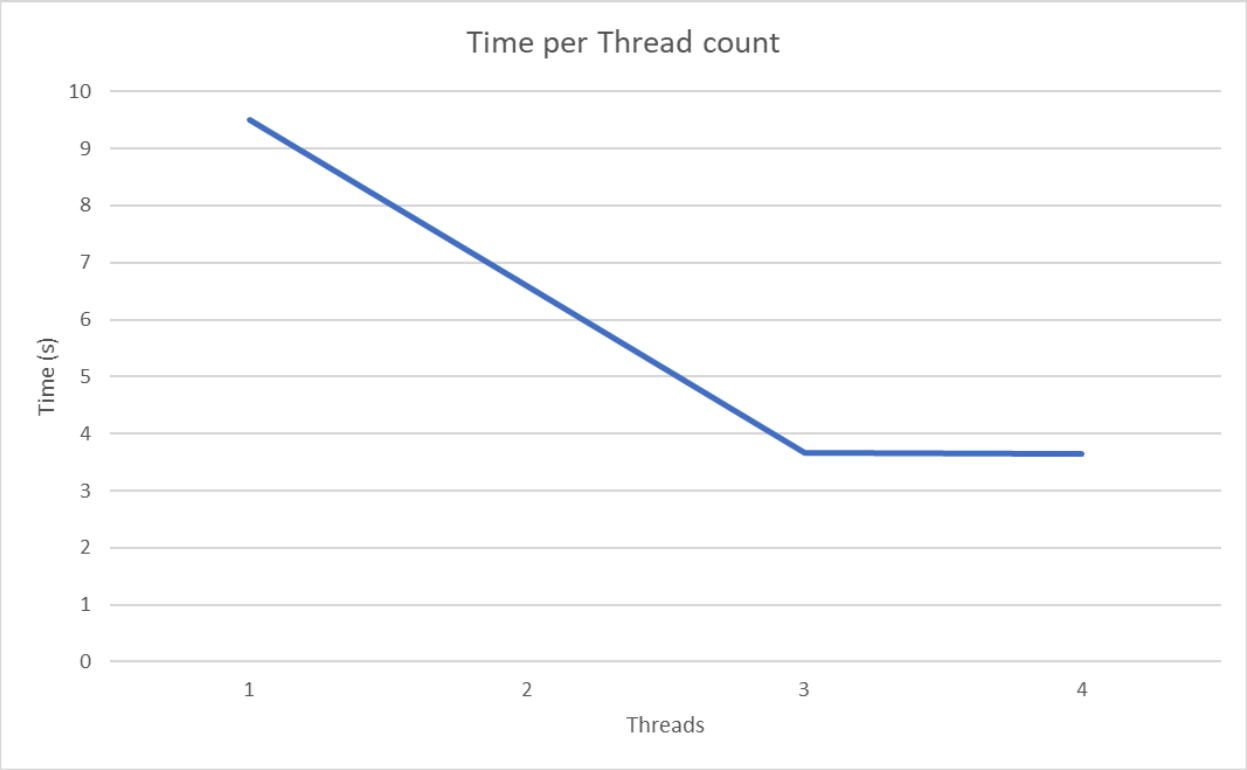
b) dct2.c emulates a 1280/960 frame, and I used the linux time function. dct2 had an average runtime of ~9.472 seconds.This translates to ~0.105 frames/second.

```
dgin@o244-01:~/Documents/csci551/numeric-parallel-starter-code/openmp_dct2$ for x in {1..5}; do
time ./dct2; done

real    0m9.508s
user    0m9.503s
sys     0m0.002s

real    0m9.448s
user    0m9.447s
sys     0m0.001s

real    0m9.507s
user    0m9.507s
sys     0m0.000s

real    0m9.451s
user    0m9.449s
sys     0m0.001s

real    0m9.447s
user    0m9.447s
sys     0m0.000s
```

c) ompdct2 had an average runtime of ~3.669. This comes out to be ~0.273 frames/second. In comparison to dct2, ompdct2 was **2.58** times faster.

```
dgin@o244-01:~/Documents/csci551/numeric-parallel-starter-code/openmp_dct2$ for x in {1..5}; do
time ./ompdct2; done

real    0m3.701s
user    0m11.049s
sys     0m0.002s

real    0m3.682s
user    0m11.060s
sys     0m0.000s

real    0m3.664s
user    0m10.989s
sys     0m0.001s

real    0m3.655s
user    0m10.978s
sys     0m0.000s

real    0m3.647s
user    0m10.957s
sys     0m0.001s
```

d) By default, ompdct2 is set to use 4 threads.Line 217 of ompdct2.c was all that needed to be altered.
4 threads took 3.654s.
3 threads took 3.667s.
2 threads took 6.599s.
1 thread took 9.512s



e) S = 4 on the cluster nodes

P = S(1-1/SU)/S-1 => 4(1-1/2.58)/4-1 => P = **0.816** parallel portion
Sp = 1 - P => **0.184** sequential portion

2)
a) In both sharpen.c and sharpen_grid.c, I changed the image dimensions to be 1260x980 instead of the default 4000x3000. I initially got the height and width mixed up, which caused the output image to have the “technicolor” effect. For repetition, I simply had the entire code repeat 5 times. I had assumed the instructions meant for this to act as an opportunity to open htop on the other window.

```
    PID USER      PRI  NI  VIRT   RES   SHR S CPU%▽MEM%   TIME+  Command
4155624 dgin       20   0  101M 12724  1872 R 187.  0.2  0:02.84 ./sharpen Alaska-Bear-1280x960.ppm Alaska-Bear-1280-sharpened.ppm
4155625 dgin       20   0  101M 12724  1872 R 17.2  0.2  0:00.26 ./sharpen Alaska-Bear-1280x960.ppm Alaska-Bear-1280-sharpened.ppm
4155628 dgin       20   0  101M 12724  1872 R 17.2  0.2  0:00.26 ./sharpen Alaska-Bear-1280x960.ppm Alaska-Bear-1280-sharpened.ppm
4155626 dgin       20   0  101M 12724  1872 R 15.2  0.2  0:00.23 ./sharpen Alaska-Bear-1280x960.ppm Alaska-Bear-1280-sharpened.ppm
4155627 dgin       20   0  101M 12724  1872 R 15.2  0.2  0:00.23 ./sharpen Alaska-Bear-1280x960.ppm Alaska-Bear-1280-sharpened.ppm
4155629 dgin       20   0  101M 12724  1872 R 15.2  0.2  0:00.23 ./sharpen Alaska-Bear-1280x960.ppm Alaska-Bear-1280-sharpened.ppm
4155633 dgin       20   0  101M 12724  1872 S 15.2  0.2  0:00.23 ./sharpen Alaska-Bear-1280x960.ppm Alaska-Bear-1280-sharpened.ppm
4155635 dgin       20   0  101M 12724  1872 S 15.2  0.2  0:00.23 ./sharpen Alaska-Bear-1280x960.ppm Alaska-Bear-1280-sharpened.ppm
4155630 dgin       20   0  101M 12724  1872 R 14.5  0.2  0:00.22 ./sharpen Alaska-Bear-1280x960.ppm Alaska-Bear-1280-sharpened.ppm
4155632 dgin       20   0  101M 12724  1872 R 14.5  0.2  0:00.22 ./sharpen Alaska-Bear-1280x960.ppm Alaska-Bear-1280-sharpened.ppm
4155634 dgin       20   0  101M 12724  1872 R 14.5  0.2  0:00.22 ./sharpen Alaska-Bear-1280x960.ppm Alaska-Bear-1280-sharpened.ppm
4155631 dgin       20   0  101M 12724  1872 R 13.9  0.2  0:00.21 ./sharpen Alaska-Bear-1280x960.ppm Alaska-Bear-1280-sharpened.ppm
   1768 root       20   0  514M 17320 13600 R  5.3  0.2  6h45:35 /opt/GC_Ext/GC/gc_linux_service
   1087 root       20   0  514M 17320 13600 S  4.6  0.2  6h53:04 /opt/GC_Ext/GC/gc_linux_service
3976246 dgin       20   0 29728 12300  6108 S  1.3  0.2  0:00.21 sshd: dgin@pts/34
4155555 dgin       20   0  9904  6104  3596 R  1.3  0.1  0:00.37 htop
```

I specified sharpen.c to utilize 12 threads for even comparison with sharpen_grid.c's 12 threads.

```
dgin@ecc-linux2:~/Documents/csci551/numeric-parallel-starter-code/openmp-sharpen$ ./sharpen Alaska-Bear-1280x960.ppm Alaska-Bear-1280-sharpened.ppm
header = P6
# Created by IrfanView
1280 960
255

START: read 0, bytesRead=0, bytesLeft=3686400
read 1, bytesRead=3686400, bytesLeft=0
END: read 1, bytesRead=3686400, bytesLeft=0

start test at 0.010439
stop test at 0.763052 for 90 frames, fps=117.947369, pps=144933726.872493

START: write 0, bytesWritten=0, bytesLeft=3686400
write 1, bytesWritten=3686400, bytesLeft=0
END: write 1, bytesWritten=3686400, bytesLeft=0
header = P6
# Created by IrfanView
1280 960
255

START: read 0, bytesRead=0, bytesLeft=3686400
read 1, bytesRead=3686400, bytesLeft=0
END: read 1, bytesRead=3686400, bytesLeft=0

start test at 0.004633
stop test at 0.797691 for 90 frames, fps=112.825644, pps=138640151.016037

START: write 0, bytesWritten=0, bytesLeft=3686400
write 1, bytesWritten=3686400, bytesLeft=0
END: write 1, bytesWritten=3686400, bytesLeft=0
header = P6
# Created by IrfanView
1280 960
255

START: read 0, bytesRead=0, bytesLeft=3686400
read 1, bytesRead=3686400, bytesLeft=0
END: read 1, bytesRead=3686400, bytesLeft=0

start test at 0.004229
stop test at 0.776676 for 90 frames, fps=115.878383, pps=142391356.813507

START: write 0, bytesWritten=0, bytesLeft=3686400
write 1, bytesWritten=3686400, bytesLeft=0
END: write 1, bytesWritten=3686400, bytesLeft=0
header = P6
# Created by IrfanView
1280 960
255

START: read 0, bytesRead=0, bytesLeft=3686400
read 1, bytesRead=3686400, bytesLeft=0
END: read 1, bytesRead=3686400, bytesLeft=0

start test at 0.004081
stop test at 0.763885 for 90 frames, fps=117.818712, pps=144775633.004295
```

Across multiple runs, the fps tried to stabilize around 120fps. However the values varied between 105fps and 123fps.
PSF convolution applies a delta function to existing values that make up an image. Enhancing edges focuses on increasing the contrast between adjacent pixel values, while blurring does the opposite.

b) For the single-threaded sharpen.c, runtime was **7.374**s. Multithreaded sharpen_grid.c had a runtime of **4.578**s.

```
START: read 0, bytesRead=0, bytesLeft=3686400
read 1, bytesRead=3686400, bytesLeft=0
END: read 1, bytesRead=3686400, bytesLeft=0

start test at 0.002081
stop test at 1.437540 for 90 frames

START: write 0, bytesWritten=0, bytesLeft=3686400
write 1, bytesWritten=3686400, bytesLeft=0
END: write 1, bytesWritten=3686400, bytesLeft=0

real    0m7.374s
user    0m7.250s
sys     0m0.028s
```

```
header = P6
# Created by IrfanView
1280 960
255


start test input at 0.000000
START: read 0, bytesRead=0, bytesLeft=3686400
read 1, bytesRead=3686400, bytesLeft=0
END: read 1, bytesRead=3686400, bytesLeft=0

completed test input at 0.001864
source file Cactus-1280x960.ppm read

start test at 0.000000

Completed test at 0.885847 for 90 create-to-join and 101.597689 FPS

Starting output file Cactus-1280-sharpenedgrid.ppm write

start test input at 0.000000
START: write 0, bytesWritten=0, bytesLeft=3686400
write 1, bytesWritten=3686400, bytesLeft=0
END: write 1, bytesWritten=3686400, bytesLeft=0

completed test input at 0.001944
Output file Cactus-1280-sharpenedgrid.ppm written

real    0m4.578s
user    0m7.686s
sys     0m0.294s
```

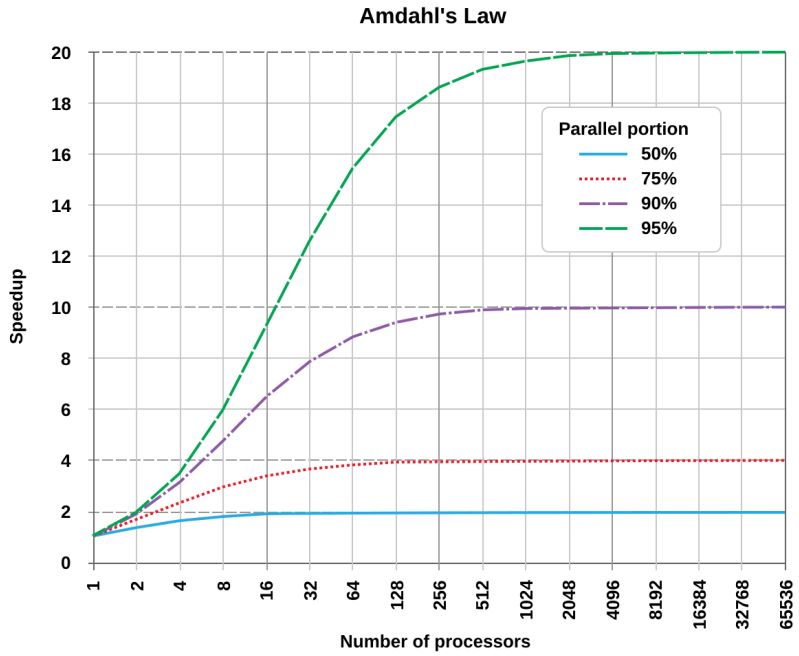
c) To compare the OpenMP sharpen.c I copied over the cactus image that the other 2 implementations tested on. OpenMP's sharpen.c runtime was even faster at **3.858**s. Compared to Pthread's sharpen_grid.c runtime of **4.578**s.

```
header = P6
# Created by IrfanView
1280 960
255

START: read 0, bytesRead=0, bytesLeft=3686400
read 1, bytesRead=3686400, bytesLeft=0
END: read 1, bytesRead=3686400, bytesLeft=0

start test at 0.004391
stop test at 0.768696 for 90 frames, fps=117.081442, pps=143869675.987761

START: write 0, bytesWritten=0, bytesLeft=3686400
write 1, bytesWritten=3686400, bytesLeft=0
END: write 1, bytesWritten=3686400, bytesLeft=0

real    0m3.858s
user    0m7.176s
sys     0m0.040s
```

d) To calculate speed up, I used SU=(T_sequential / T_parallel).

(single thread / Pthread) 7.374 / 4.578 = **1.612** times faster

(single thread / OpenMP) 7.374 / 3.858 = **1.911** times faster



ECC has 2 cores(or for the sake of the graph's wording, processors), so a speedup of **1.911** fits nearly perfectly on the graph. Pthread's **1.612** times speedup seems to imply a lower parallel portion value.

The assignment1.zip contains all of the code that I used. All 3 sharpen files are in the assignment1 directory. The dct2 folder remained separate because it was run on the cluster. For running code, run the Makefile in the respective directory. The command line arguments are the same as initially given: ./[executable name] [input image.ppm] [output image.ppm]

```
dgin@ecc-linux2:~/Documents/csci551/assignment1$ ./sharpen Alaska-Bear-1280x960.ppm Alaska-Bear-1280-sharpened.ppm
header = P6
# Created by IrfanView
1280 960
255

START: read 0, bytesRead=0, bytesLeft=3686400
read 1, bytesRead=3686400, bytesLeft=0
END: read 1, bytesRead=3686400, bytesLeft=0

start test at 0.014120
stop test at 1.477340 for 90 frames, fps=60.920295, pps=74858858.339208

START: write 0, bytesWritten=0, bytesLeft=3686400
write 1, bytesWritten=3686400, bytesLeft=0
END: write 1, bytesWritten=3686400, bytesLeft=0
```