Final Parallel Program

Image Rotation

1) **Final program description**

   The major goal and objective of this program is to speed up large vector manipulation to rotate pgm images.

2) **The value of the solution**

   The value of this solution is that this is used for image processing and vector manipulation is a very intensive task and without parallelism video processing would not be possible

3) **What numerical methods and algorithms are used and what type of math is required**

   Numerical methods: Vector operations and transformations

   Algorithms/math: 2-D convolution and transformation

Improvement and Refactoring Examples (Check Used Column)

Numerical Method	Mathematics	Description	Used
Vector/matrix operations and transformations	Convolution and transformation	Use of 2-D convolution and transformation functions applied to images (e.g. DCT, rotation, image sharpen/blur, etc.)	X
Prime number searching and testing	Prime number theorem	Use of Sieve of Eratosthenes and more advanced methods to find prime density, largest prime in range and list prime numbers in an interval	
Integration	Calculus	Use of Riemann sums, Trapezoidal, Simpson's Rule, or advanced Runge-Kutta	
Non-linear function generation (and integration)	Calculus, Accuracy and Precision	Integration of non-linear functions and sources of error compared to definite integrals	
Gaussian Elimination with	Linear systems	Solving systems of equations that describe linear systems (circuits, fluid flow and	

4) **What Parallel programming methods were used**

   OpenMP

Parallel Programming Methods Used (Check mark Used Column)

Parallel Programming Method	Description	Used
POSIX threads	Shared memory threading within a Linux process	
MPI	Message Passing Interfaces between Linux processes on the same node or network interconnected nodes	
OpenMP	Compiler directives to generate parallel shared memory code for specific parts of a program	X
Other	CUDA, OpenCL, hybrid combination, etc.	
Description Please describe methods you used here.		

2) Sequential solution and computation time.

1) Sequential program

Code will be provided in submission with makefile and source file

2)Timing

```
juan@DESKTOP-QCQU6MF:$ time ./rotation right
Time program took after File I/O: 4.65265s

real    0m4.682s
user    0m1.597s
sys     0m0.238s
juan@DESKTOP-QCQU6MF:$ time ./rotation right
Time program took after File I/O: 3.86093s

real    0m3.902s
user    0m1.391s
sys     0m0.222s
juan@DESKTOP-QCQU6MF:$ time ./rotation right
Time program took after File I/O: 4.01162s

real    0m4.054s
user    0m1.897s
sys     0m0.167s
juan@DESKTOP-QCQU6MF:$ time ./rotation right
Time program took after File I/O: 3.65962s

real    0m3.690s
user    0m1.320s
sys     0m0.247s
juan@DESKTOP-QCQU6MF:$ time ./rotation right
Time program took after File I/O: 4.49412s

real    0m4.517s
user    0m1.397s
sys     0m0.257s
juan@DESKTOP-QCQU6MF:$ time ./rotation right
Time program took after File I/O: 4.25669s

real    0m4.291s
user    0m1.706s
sys     0m0.217s
juan@DESKTOP-QCQU6MF:$ time ./rotation right
Time program took after File I/O: 4.25564s

real    0m4.279s
user    0m1.638s
sys     0m0.185s
juan@DESKTOP-QCQU6MF:$ time ./rotation right
Time program took after File I/O: 4.56839s

real    0m4.629s
user    0m1.645s
sys     0m0.271s
juan@DESKTOP-QCQU6MF:$ time ./rotation right
Time program took after File I/O: 3.51991s

real    0m3.560s
user    0m1.347s
sys     0m0.209s
juan@DESKTOP-QCQU6MF:$ time ./rotation right
Time program took after File I/O: 3.61296s

real    0m3.651s
user    0m1.359s
sys     0m0.193s
juan@DESKTOP-QCQU6MF:$
```

$$\frac{(4.682 + 3.902 + 4.054 + 3.690 + 4.517 + 4.291 + 4.279 + 4}{10} = 4.1255$$

Here are 10 trials and the average time of all these runs is 4.1255s

3) Parallel design and solution with computation time.

1)parallel program with makefile and source code will be providedin submition

2)Timing

```
juan@DESKTOP-QCQU6MF:$ time ./omp_rotation right
Time program took after File I/O: 0.718463s

real    0m0.755s
user    0m3.749s
sys     0m2.979s
juan@DESKTOP-QCQU6MF:$ time ./omp_rotation right
Time program took after File I/O: 0.720972s

real    0m0.764s
user    0m4.724s
sys     0m2.958s
juan@DESKTOP-QCQU6MF:$ time ./omp_rotation right
Time program took after File I/O: 0.752515s

real    0m0.784s
user    0m4.634s
sys     0m2.629s
juan@DESKTOP-QCQU6MF:$ time ./omp_rotation right
Time program took after File I/O: 0.697687s

real    0m0.751s
user    0m4.124s
sys     0m2.872s
juan@DESKTOP-QCQU6MF:$ time ./omp_rotation right
Time program took after File I/O: 0.723925s

real    0m0.769s
user    0m4.192s
sys     0m2.922s
juan@DESKTOP-QCQU6MF:$ time ./omp_rotation right
Time program took after File I/O: 0.701066s

real    0m0.739s
user    0m4.050s
sys     0m2.896s
juan@DESKTOP-QCQU6MF:$ time ./omp_rotation right
Time program took after File I/O: 0.707313s

real    0m0.746s
user    0m4.075s
sys     0m2.942s
juan@DESKTOP-QCQU6MF:$ time ./omp_rotation right
Time program took after File I/O: 0.820943s

real    0m0.866s
user    0m4.468s
sys     0m2.703s
juan@DESKTOP-QCQU6MF:$ time ./omp_rotation right
Time program took after File I/O: 0.687443s

real    0m0.724s
user    0m3.991s
sys     0m2.859s
juan@DESKTOP-QCQU6MF:$ time ./omp_rotation right
Time program took after File I/O: 0.718191s

real    0m0.759s
user    0m4.330s
sys     0m2.971s
juan@DESKTOP-QCQU6MF:$
```

$$\frac{(0.755 + .764 + .784 + .751 + .769 + .739 + .746 + .866 + .7}{10} = 0.7657$$

Average time for the parallel run time is 0.7657

4) Parallel speed-up analysis comparing results to Amdahl's law.

1) Determine parameters for Amdahl's Law:

Amdahl's Law parameter	How obtained?	Description
Sequential portion (% of total)	1-P	6% Sequential
Parallel portion (% of total)	Parallel Time/Total Time	94% Parallel
Number of shared memory cores used and type	My laptop has 16 cores available and i used htop to verify if all cores are being used	16 cores clocked at 2.5GHz
Number of nodes used in MPI distributed program X # of cores per node	Not using MPI	N/A
Final value used for S, the scaling factor	16	I used all 16 cores according to htop

Parallel portion of the code can be determined by taking a sample run of our program and dividing the time the parallel portion of the program took by the total time the program took to run

```
juan@DESKTOP-QCQU6MF:$ time ./omp_rotation right
Time program took after File I/O: 0.718191s

real    0m0.759s
user    0m4.330s
sys     0m2.971s
```

P= .718/.759 = .94

Sequential = 1-.94. = ~~.05~~ .06

# of cores used = 16 clocked at 2.5GHz

```
1  [||||||     20.0%]  5  [||||     16.2%]  9  [||||||    23.4%]  13 [|||||    17.9%]
2  [||||       16.3%]  6  [||||     16.3%]  10 [||||      15.2%]  14 [||||     17.3%]
3  [|||||      19.0%]  7  [||||     15.1%]  11 [||||      15.1%]  15 [|||||    16.7%]
4  [|||||      18.4%]  8  [|||||    16.2%]  12 [|||||     16.8%]  16 [|||||    18.4%]
```

2) Plot Amdahl's Law ideal speed-up and your actual speed-up

Point 1:

```
juan@DESKTOP-QCQU6MF:$ time ./omp_rotation right
Time program took after File I/O: 0.552776s

real    0m0.573s
user    0m3.007s
sys     0m2.097s
juan@DESKTOP-QCQU6MF:$ 
```

P = .552/.573 = .96

Sequential = 1-.96 = .03

#of cores used 16 clocked at 2.5GHz

Ideal speed up 25x

$P = .96$	$= 0.96$
$\frac{1}{(1-P)+0}$	$= 25$

Actual speed up 10x

$\frac{1}{(1-P)+\frac{P}{S}}$	$= 10$
$S = 16$	$= 16$
$P = .96$	$= 0.96$

Point 2:

```
juan@DESKTOP-QCQU6MF:$ time ./omp_rotation right
Time program took after File I/O: 0.570141s

real    0m0.592s
user    0m2.900s
sys     0m2.167s
```

P = .570/.592 = .96

Sequential = 1-.96 = .03

#of cores used 16 clocked at 2.5GHz

Ideal speed up 25x

$$P = .96 \qquad = 0.96$$

$$\frac{1}{(1-P)+0} \qquad = 25$$

Actual speed up 10x

$$\frac{1}{(1-P)+\frac{P}{S}} \qquad = 10$$

$$S = 16 \qquad = 16$$

$$P = .96 \qquad = 0.96$$

Point 3:

```
juan@DESKTOP-QCQU6MF:$ time ./omp_rotation right
Time program took after File I/O: 0.525898s

real    0m0.551s
user    0m2.984s
sys     0m2.044s
```

P = .525/.551 = .95

Sequential = 1-.95 = .04

#of cores used 16 clocked at 2.5GHz

Ideal Speed up 20x

$$P = .95 \qquad = 0.95$$

$$\frac{1}{(1-P)+0} \qquad = 20$$

Actual Speed up 9.1x

$$\frac{1}{(1-P)+\frac{P}{S}} \qquad = 9.142857143$$

$$S = 16 \qquad = 16$$

$$P = .95 \qquad = 0.95$$