JONATHAN PERRY
4/7/18

PROJECT #0

For this assignment, I created a bash script to help test my program with several different combinations of parameters – such as the number of threads used, and the input size of the array being worked on. The bash script runs the program with each input size separately with 1 thread set, then repeats the same process over, but with 4 threads set. To run the script, simply enter in the following command: **./script**

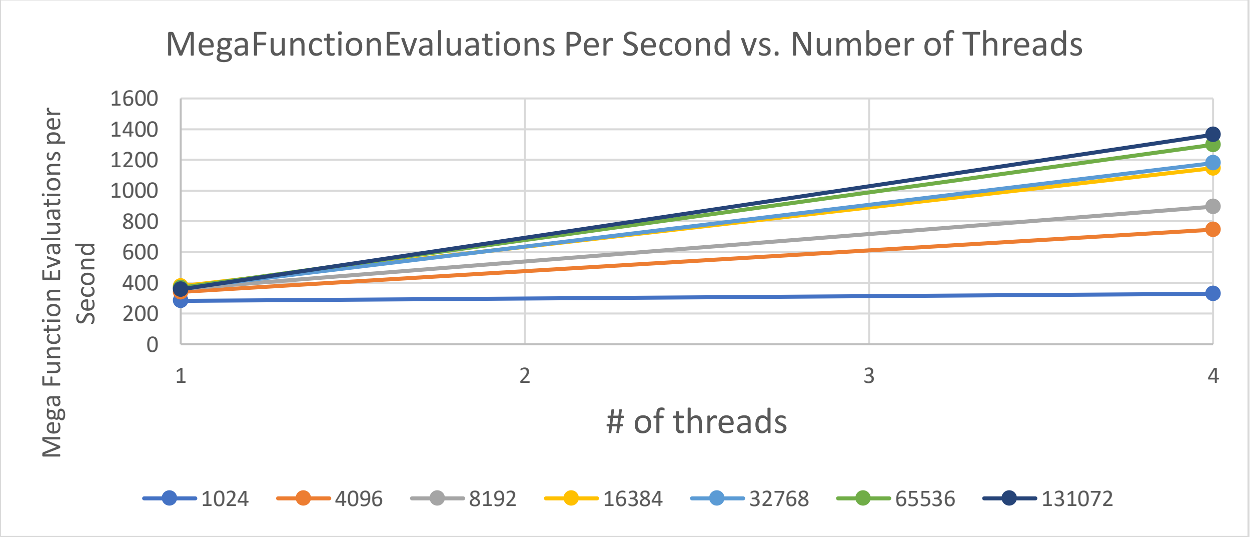
1. Tell what machine you ran this on

   - A Linux system on OSU's FLIP server was used to run Project #0.

2. What performance results did you get?

   - **Performance** (Mega Function Evaluations Per Second)

| *# of threads* | *1* | *4* |
|---|---|---|
| *Input size: 1024* | 282.55 | 328.59 |
| *Input size: 4096* | 340.84 | 746.16 |
| *Input size: 8192* | 360.47 | 895.96 |
| *Input size: 16384* | 378.33 | 1146.54 |
| *Input size: 32768* | 364.26 | 1179.18 |
| *Input size: 65536* | 368.15 | 1298.59 |
| *Input size: 131072* | 357.68 | 1363.7 |



   - **Elapsed Time** (seconds)

| *# of threads* | *1* | *4* |
|---|---|---|
| *Input size: 1024* | 0.00000297 | 0.0000026 |
| *Input size: 4096* | 0.00001145 | 0.00000527 |

| | | |
|---|---|---|
| *Input size: 8192* | 0.00002202 | 0.00000772 |
| *Input size: 16384* | 0.00004174 | 0.00001392 |
| *Input size: 32768* | 0.00008511 | 0.00002481 |
| *Input size: 65536* | 0.00016656 | 0.00004545 |
| *Input size: 131072* | 0.000381 | 0.000105 |

3. What was your 4-thread-to-one-thread speedup?

| *4-to-1-thread speedup(sec)* | |
|---|---|
| *Input size: 1024* | 1.142307692 |
| *Input size: 4096* | 2.17267552 |
| *Input size: 8192* | 2.85233161 |
| *Input size: 16384* | 2.99856322 |
| *Input size: 32768* | 3.43047158 |
| *Input size: 65536* | 3.66468647 |
| *Input size: 131072* | 3.641349 |

4. Why do you think it is behaving this way?

The 4-to-1-thread speedup is behaving this way because OpenMP allows your program to break up the work (of multiplying 2 large floating-point arrays) into separate threads – which all run simultaneously. The speedup increases as the input size increases due to the increased performance gained from using 4 threads, versus 1 thread. From using a higher number of threads, less time is being taken up to do the same amount of work.

5. What was your Parallel Fraction, Fp?

| *Parallel Fraction* | |
|---|---|
| *Input size: 1024* | 0.166105499 |
| *Input size: 4096* | 0.71965066 |
| *Input size: 8192* | 0.8658795 |
| *Input size: 16384* | 0.88867593 |
| *Input size: 32768* | 0.94465985 |
| *Input size: 65536* | 0.96950048 |
| *Input size: 131072* | 0.967169 |