Test Methodology

To ensure we are getting significant and meaningful runtime values, we use an adequately large sample size to perform FFT: 222 samples or 4,194,304 samples. In order to generate complex waveform to perform FFT on, we generate three cosine waves. The first waveform has frequency 1 and amplitude 10. The second waveform has frequency 2 and amplitude 4. The third waveform has frequency 4 and amplitude 5. We generate the samples over 2 \* π, and the summed waveform is used as the input for FFT calculation. To test correctness, we perform FFT and get correct resulting FFT graph with the correct frequency values and relative amplitudes in the input waves. We will take the runtime average of ten runs, and use the average values to normalize runtime fluctuations, and make calculations on the same machine so that runtimes are can be compared fairly. We will use the same sample size to test the recursive serial Cooley-Tukey FFT algorithm, the improved serial iterative algorithm, and the parallelized iterative algorithm using openMP. We will then compare the runtimes between all algorithms, and analyze the scalability and efficiency of the parallelized openMP algorithm.

Figure x: openMP efficiency and speed-up

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| p | 1 | 2 | 4 | 8 |
| S | 1.0 | 1.5669 | 2.1580 | 3.7656 |
| E | 1 | 0.7835 | 0.5395 | 0.4710 |

Figure x: openMP speedup scatter chart

Figure x: openMP efficiency scatter chart

Figure x: Serial optimization

Figure x: openMP runtime vs threads

Figure x: Samples of input waves

Figure x: Sum of waves for transform

Figure x: FFT result