

# FFT Algorithm Comparison for GPUs and CPUs

Jaden Meyer and Jayden Burr

November 28, 2024

# FFT Algorithm

This is some text in the first frame. This is some text in the first frame. This is some text in the first frame.

# GPU Speeds

The FFT Algorithm has time complexity of  $O(N \log N)$ . The average GPU has hundreds to tens of thousands of cores. It can compute things in parallel.

Using an average graphics card we expect 4-8 TFLOPs ( $10^{12}$  operations):

# GPU Speeds

The FFT Algorithm has time complexity of  $O(N \log N)$ . The average GPU has hundreds to tens of thousands of cores. It can compute things in parallel.

Using an average graphics card we expect 4-8 TFLOPs ( $10^{12}$  operations):

For a DFT with  $N = 1\text{million}$ :  $(10^6)^2 \cdot \frac{1}{4 \cdot 10^{12}} = 0.25$  seconds

# GPU Speeds

The FFT Algorithm has time complexity of  $O(N \log N)$ . The average GPU has hundreds to tens of thousands of cores. It can compute things in parallel.

Using an average graphics card we expect 4-8 TFLOPs ( $10^{12}$  operations):

For a DFT with  $N = 1\text{million}$ :  $(10^6)^2 \cdot \frac{1}{4 \cdot 10^{12}} = 0.25$  seconds

For a FFT with  $N = 1\text{million}$ :  $(10^6) \cdot \log_2(10^6) \cdot \frac{1}{4 \cdot 10^{12}} = 0.000005$  seconds

# GPU Speeds

The FFT Algorithm has time complexity of  $O(N \log N)$ . The average GPU has hundreds to tens of thousands of cores. It can compute things in parallel.

Using an average graphics card we expect 4-8 TFLOPs ( $10^{12}$  operations):

For a DFT with  $N = 1\text{million}$ :  $(10^6)^2 \cdot \frac{1}{4 \cdot 10^{12}} = 0.25$  seconds

For a FFT with  $N = 1\text{million}$ :  $(10^6) \cdot \log_2(10^6) \cdot \frac{1}{4 \cdot 10^{12}} = 0.000005$  seconds

That is FAST...

# CPU Speeds

A CPU has only a handful of cores. It does not use parallelization and is very weak at 'indirect' computations. Occasionally the core will use 1 or 2 threads.

# CPU Speeds

A CPU has only a handful of cores. It does not use parallelization and is very weak at 'indirect' computations. Occasionally the core will use 1 or 2 threads.

Most modern CPUs use around 4-5GHz which means a single-core operation speed of: 0.02 TFLOPs:

For a DFT with  $N = 1\text{million}$ :  $(10^6)^2 \cdot \frac{1}{0.02 \cdot 10^{12}} = 50$  seconds



# CPU Speeds

A CPU has only a handful of cores. It does not use parallelization and is very weak at 'indirect' computations. Occasionally the core will use 1 or 2 threads.

Most modern CPUs use around 4-5GHz which means a single-core operation speed of: 0.02 TFLOPs:

For a DFT with  $N = 1\text{million}$ :  $(10^6)^2 \cdot \frac{1}{0.02 \cdot 10^{12}} = 50$  seconds

For a FFT with  $N = 1\text{million}$ :  $(10^6) \cdot \log_2(10^6) \cdot \frac{1}{0.02 \cdot 10^{12}} = 0.001$  seconds

# CPU Speeds

A CPU has only a handful of cores. It does not use parallelization and is very weak at 'indirect' computations. Occasionally the core will use 1 or 2 threads.

Most modern CPUs use around 4-5GHz which means a single-core operation speed of: 0.02 TFLOPs:

For a DFT with  $N = 1\text{million}$ :  $(10^6)^2 \cdot \frac{1}{0.02 \cdot 10^{12}} = 50$  seconds

For a FFT with  $N = 1\text{million}$ :  $(10^6) \cdot \log_2(10^6) \cdot \frac{1}{0.02 \cdot 10^{12}} = 0.001$  seconds

That is quite slow...

# Results?