CUDA Final Project Proposal

Wen-Liang, Huang 黃文亮

May 20, 2020

1 2D & 3D FFT with GPU

We have already known the 1D fast Fourier transform[1] with GPU in tutorial and the professor's code, and we also know that there are fundational library base on the Cooley-Tuckey[2] and Bluestein algorithms, and time result in GPU is better than CPU fast fourier transform. However; although I have read the three dimension FFT on GPU [3], I still want to run the 2D FFT and 3D FFT in CPU and GPU, then compare the result of this two different types.

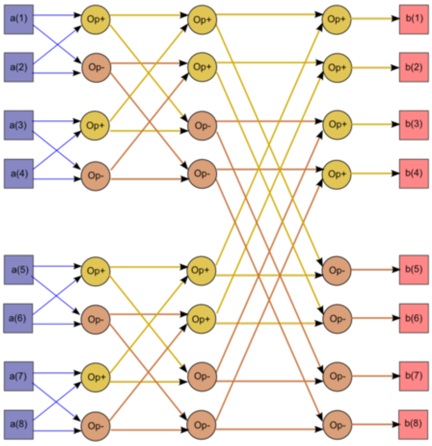
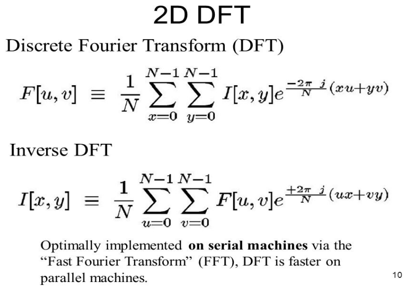


Figure 1: Simple Cooley-Tuckey Algorithm

Next, I will quickly explain that how can we use the 2D and 3D fast Fourier transform.

1.1 2D FFT

In the 2D FFT, I have to check the 2D discrete Fourier transform and inverse discrete Fourier Transform Actually, the 2D FFT is extended by 1D FFT.The concept is that I have to use the 1D FFT in each column at first, then using the transform result to put them into each row.



2D DFT

Discrete Fourier Transform (DFT)

$$F[u,v] \equiv \frac{1}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} I[x,y]e^{\frac{-2\pi}{N}j(xu+yv)}$$

Inverse DFT

$$I[x,y] \equiv \frac{1}{N}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1} F[u,v]e^{\frac{+2\pi}{N}j(ux+vy)}$$

Optimally implemented **on serial machines** via the "Fast Fourier Transform" (FFT), DFT is faster on parallel machines.

10

Figure 2: 2D DFT

1.2 3D FFT

There is no suitable 3D FFT coding, but I think it just add another one dimension from 2D FFT, but remember, I have to run each column, then use the result of each column to put into each row, then use the result of each row to put into the next dimension.

2 Expected Result

There are two temporary expected result in my proposal

2.1 Run time in 2D & 3D FFT

Compared the runtime of 2D and 3D FFT with GPU to the runtime of 2D and 3D FFT with CPU, I expect that the run time of the previous one is less than the later one.

2.2 Other Application with FFT

Using the library of FFT in Nvidia and implement some social application.

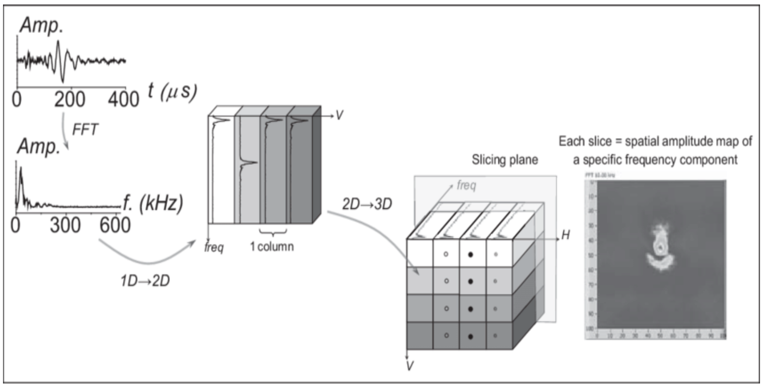


Figure 3: FFT of 1D to 2D to 3D

References

[1] Fast Fourier Transform. https://en.wikipedia.org/wiki/Fast_Fourier_transform.

[2] Cooley-Tuckey Algorigthm. https://en.wikipedia.org/wiki/Cooley-Tukey_FFT_algorithm.

[3] Selcuk Keskin, Ertunc Erdil, and T. Kocak. An efficient parallel implementation of 3 d-fft on gpu. 2017.