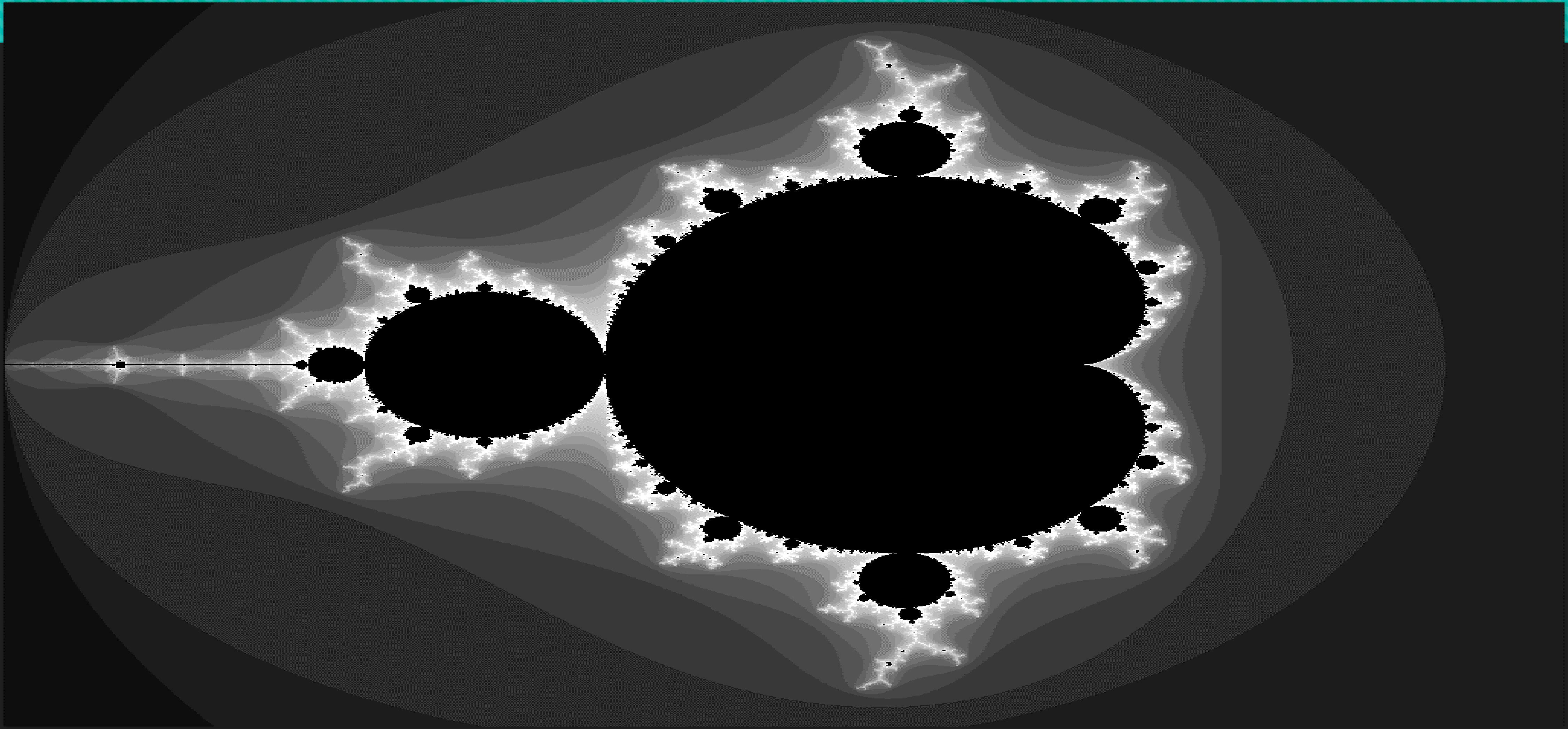
MANDELBROT SET USING CUDA

By Sanjay Muthu Manickam

Mandelbrot Set

The **Mandelbrot set** is the set of complex numbers c for which the function $f_c(z) = z^2 + c$ does not diverge when iterated from $z = 0$, *i.e* for which the function $f_c(0), f_c(f_c(0))$ *etc*. remain bounded in absolute value.

Fractal Generated Using CUDA



How it works

- To determine whether a value is in or out of the Mandelbrot Set , we loop through the formula $Z_{n+1} = Z^2_n + C$ a certain number of times , these iteration number determines the color of the pixel whether it is black or white.
- The above equation basically translates to this $\lim_{n\text{->}\infty} Z_n \leq 2$

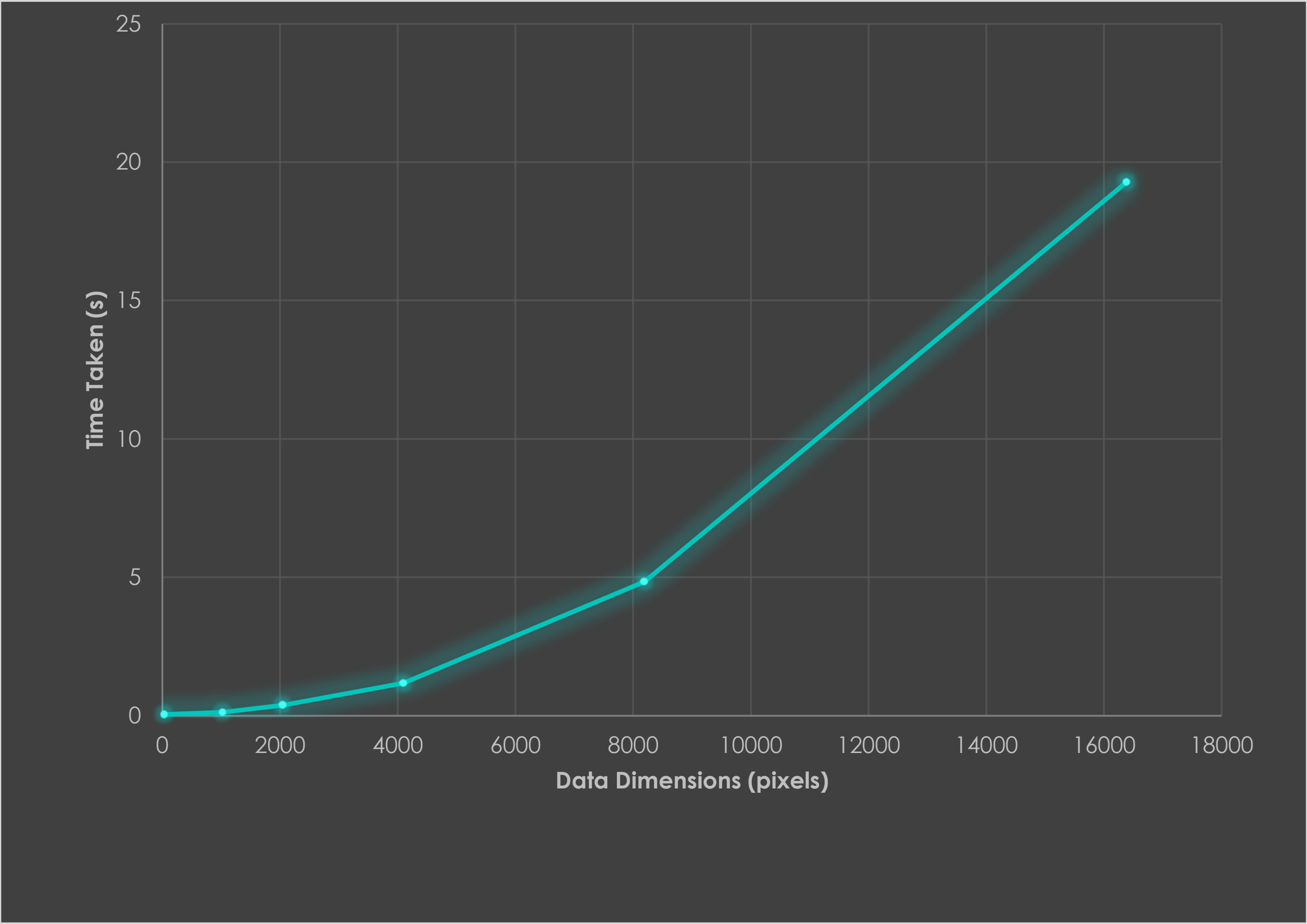
Work Flow

- Assign space for the image and load it into the GPU.
- Launch Kernel.
- Assign a grid and block size so that the image is divided across the CUDA cores.
- Copy the image back to the host.
- Write it into a bmp image

Challenges in using CUDA

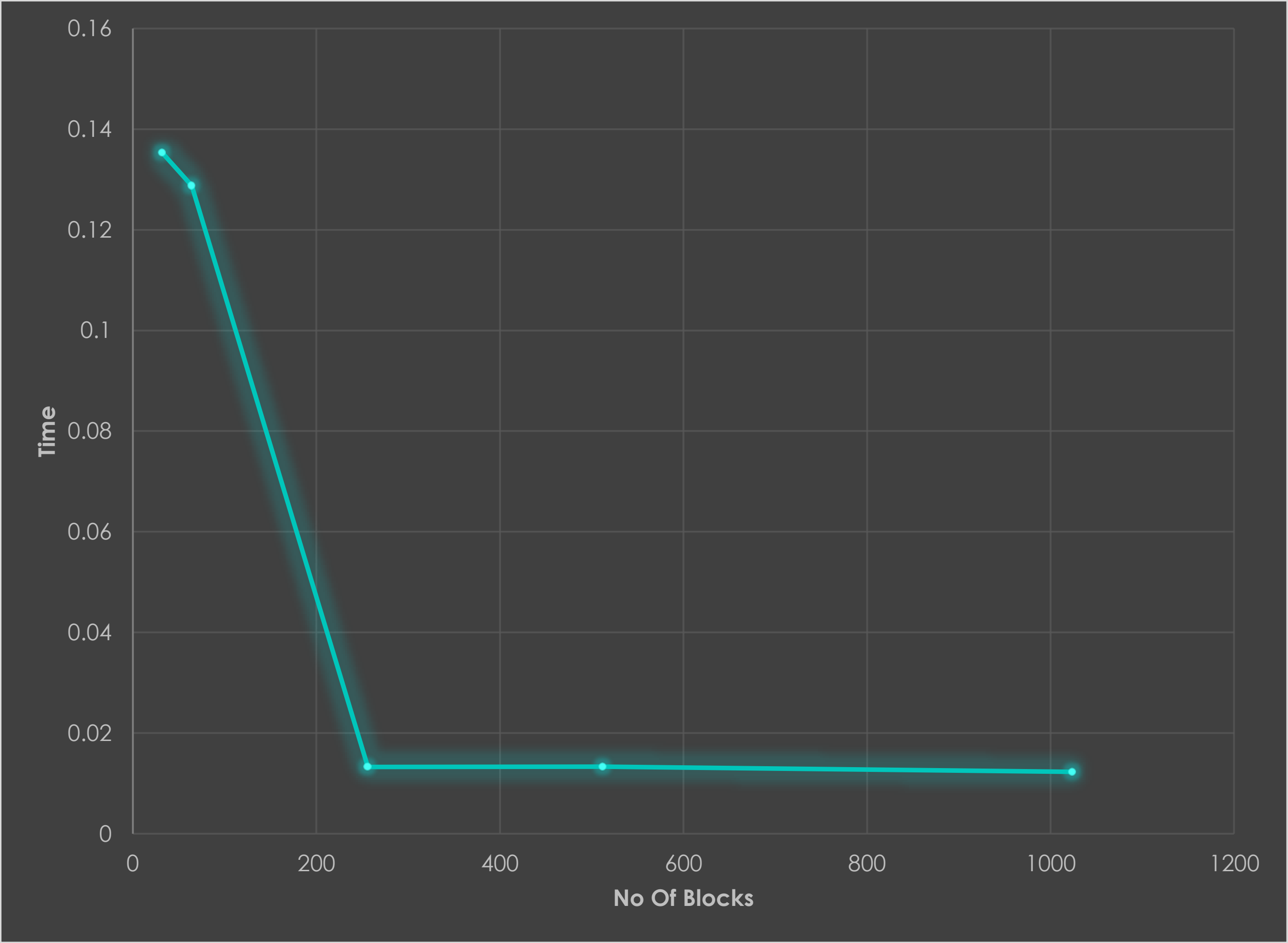
- The challenge in CUDA programming is choosing the right grid and block size.
- A block with many threads will run faster than many blocks with one thread.
- This is because of the bottleneck of how many warps can be executed concurrently
- So for best results it is better to use the no of threads per block a multiple of 32.

Image Size vs No of Threads



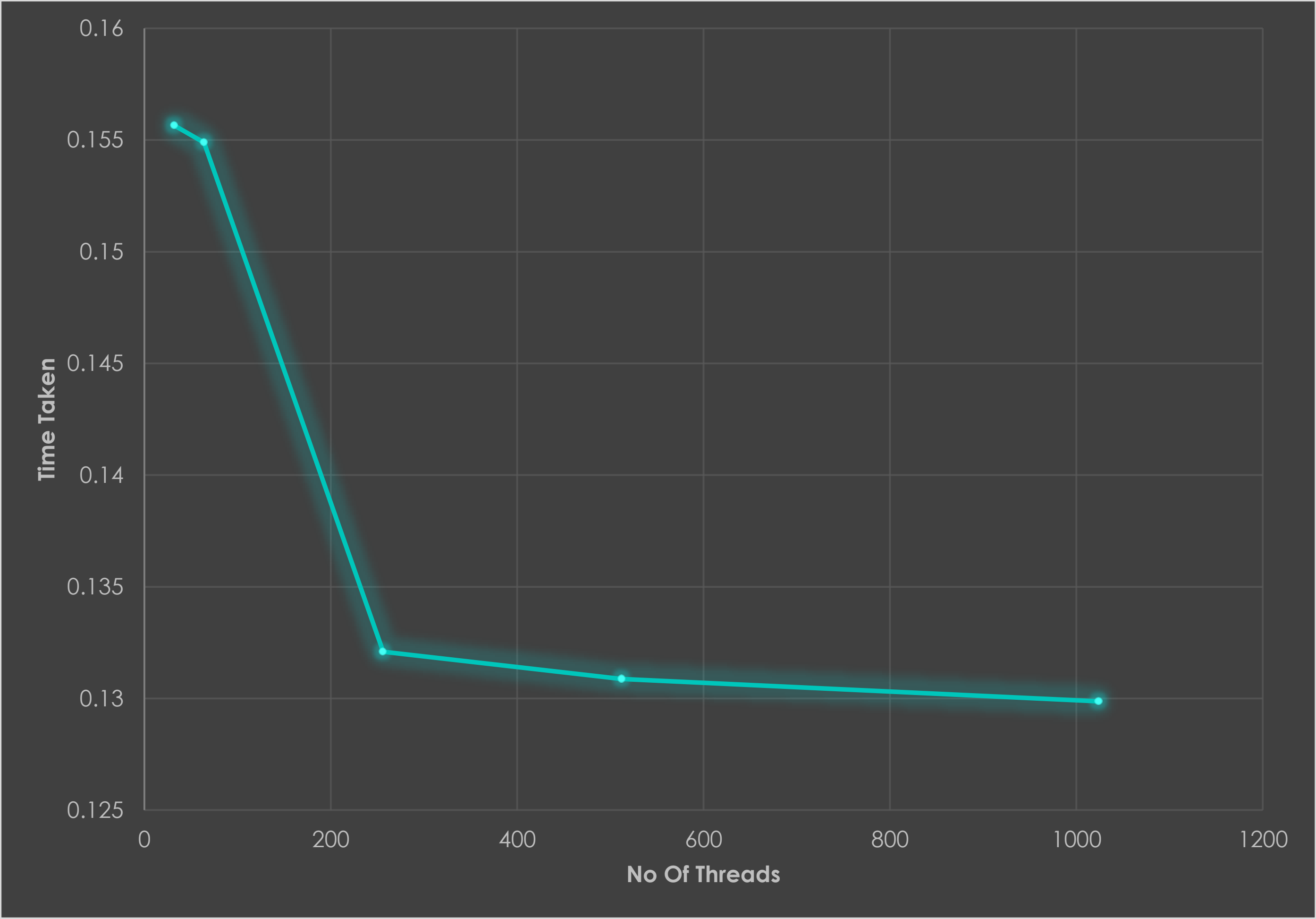
Data Dimensions	Time Taken(s)
32	0.046023
1024	0.122342
2048	0.382976
4096	1.176499
8192	4.838946
16384	19.275684

Single Thread , Multiple Blocks



No of Blocks	Time Taken(s)
32	0.1353
64	0.1287
256	0.0163
512	0.0133
1024	0.0123

Multiple Blocks , Single Thread



No of Blocks	Time Taken
32	0.155648
64	0.154892
256	0.132096
512	0.130874
1024	0.129874

THANK YOU