

Cuda

Abe Wiersma, Bram van den Akker

December 8, 2013

1 Wave equation with cuda

Because we're unable to complete part 1 of assignment 3 the results are only compared to assignment 1 and 2.

In figure 1 you can clearly see that cuda performance is much higher than cpu multithreading when using large datasets. Where Pthreads and openMP need a lot of time to calculate the wave with an i of 10^7 cuda doesn't have much trouble getting fast results.

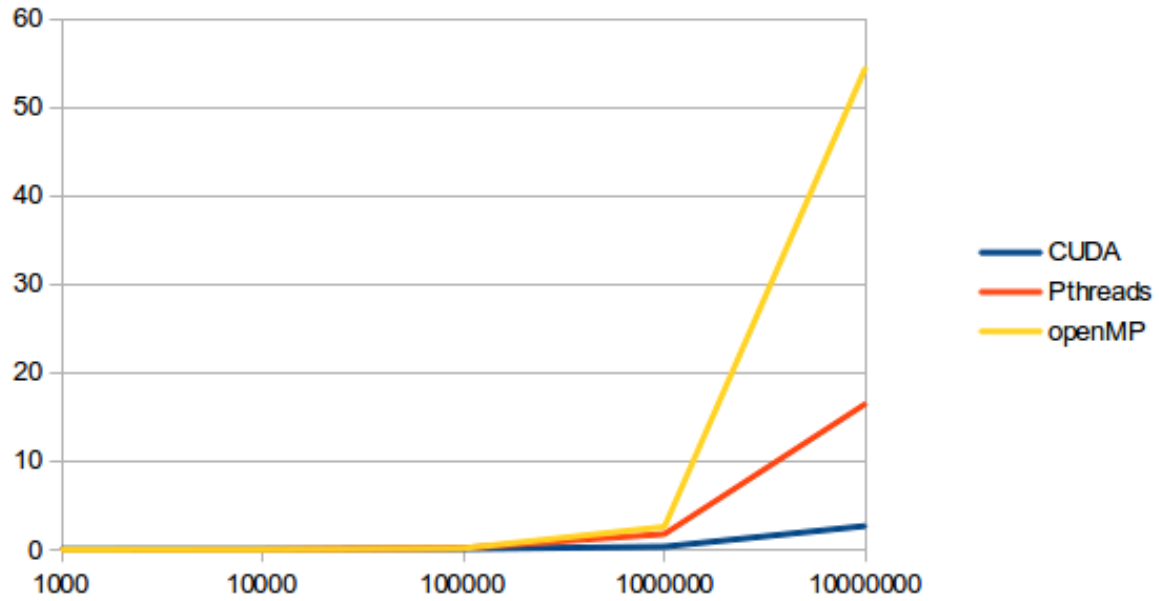


Figure 1: Cuda vs pthreads and openmp. Cuda block size is 512, pthreads and openmp both use 8 cores.

1.1 Blocksizes

As you can see in the graph below a higher block size does not directly relate to a higher performance. When using block sizes larger than the amount of cores available the performance will go down (a GTX-480 has 448 cuda cores, this is why 512 is slower than 256 cores). Small blocksizes (i.e. 8, 16 and 32) will exceed the maximum amount of blocks when trying to run large datasets and fail.

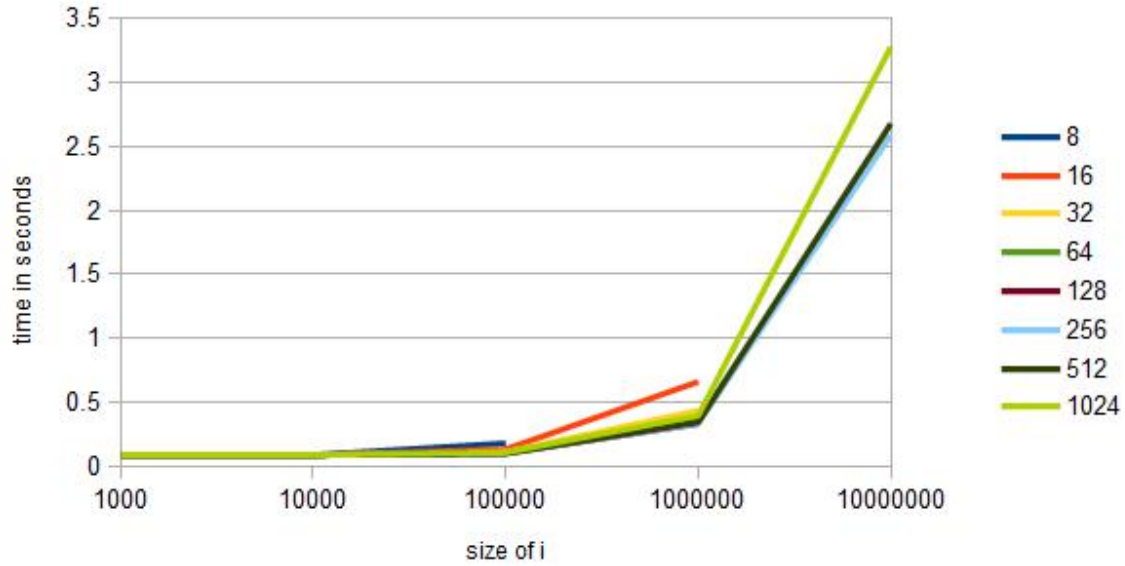


Figure 2: Cuda scheduling with different block sizes.

1.2 Result comparison

When using a small i the results of the cuda implementation are really inaccurate, increasing the i value will get much better results. This could be due to the use of floats instead of doubles. Float performance on cuda machines is much better but will, when dealing with small values lose a lot of precision. figure 4 through 8 show the cuda and pthreads results side by side.

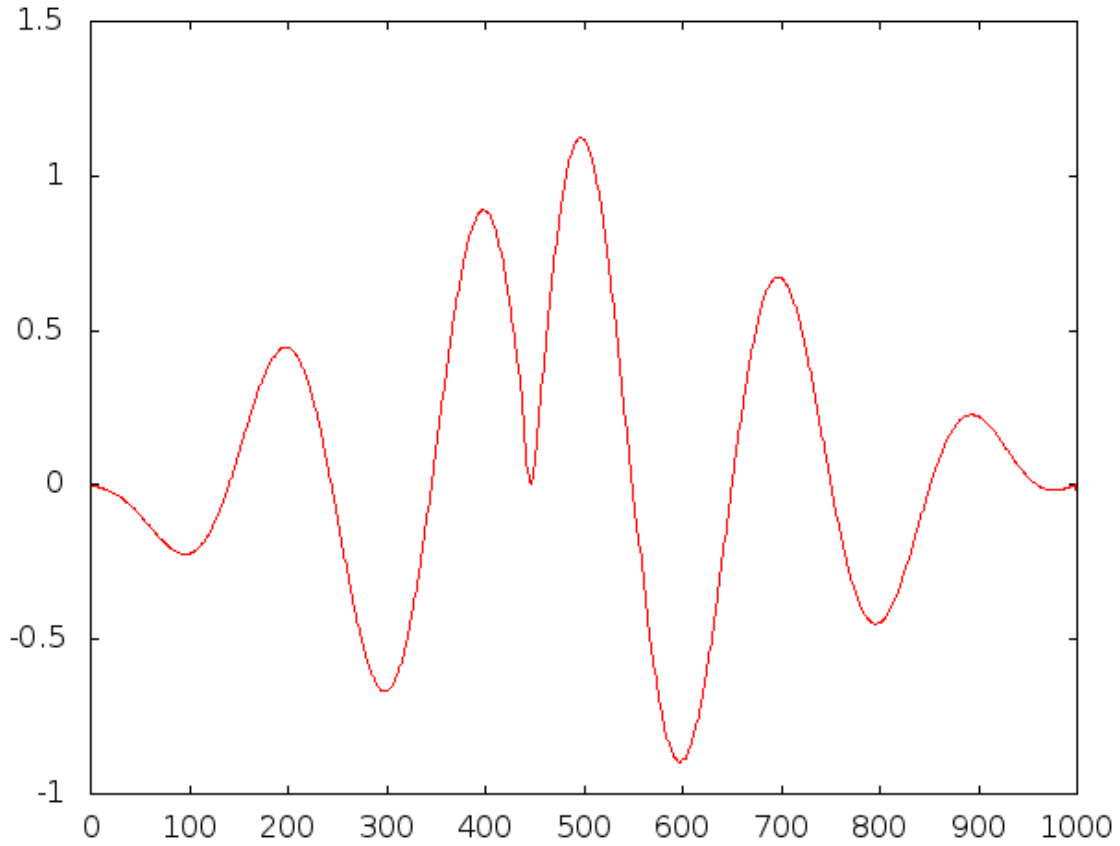


Figure 3: Correct result of $i = 1000$.

2 Finding the max value

When running both a cpu and cuda algorithm to find the maximum value of an array of floats one thing is really important: Array length. With a small amount of float values the CPU algorithm will perform better then the cuda algorithm. When the amount of float values is increased the cuda algorithm starts out performing the cpu algorithm.

N floats	cpu	cuda
10000	0.044096 <i>ms</i>	0.244032 <i>ms</i>
1000000	4.19827 <i>ms</i>	1.60947 <i>ms</i>
100000000	419.26 <i>ms</i>	117.667 <i>ms</i>

The performance of the cpu algorithm vs the cuda algorithm.

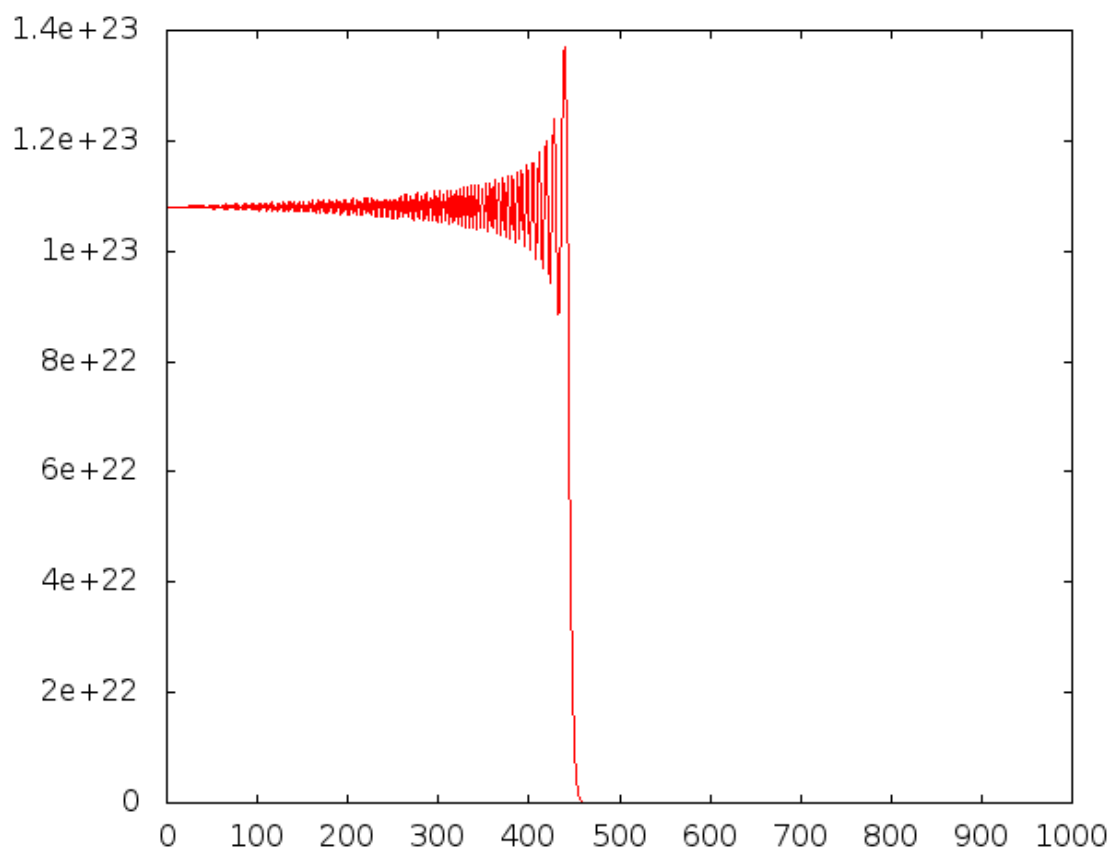


Figure 4: Cuda result of $i = 1000$.

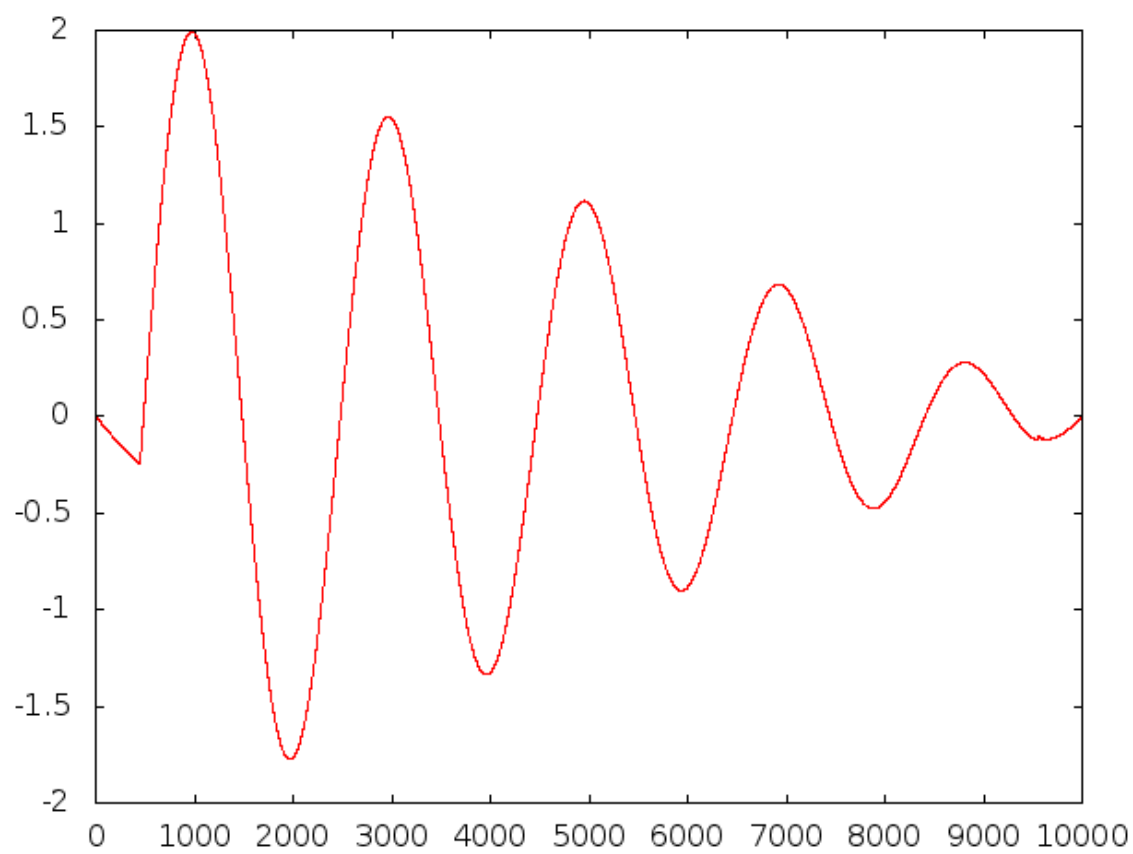


Figure 5: Correct result of $i = 10000$.

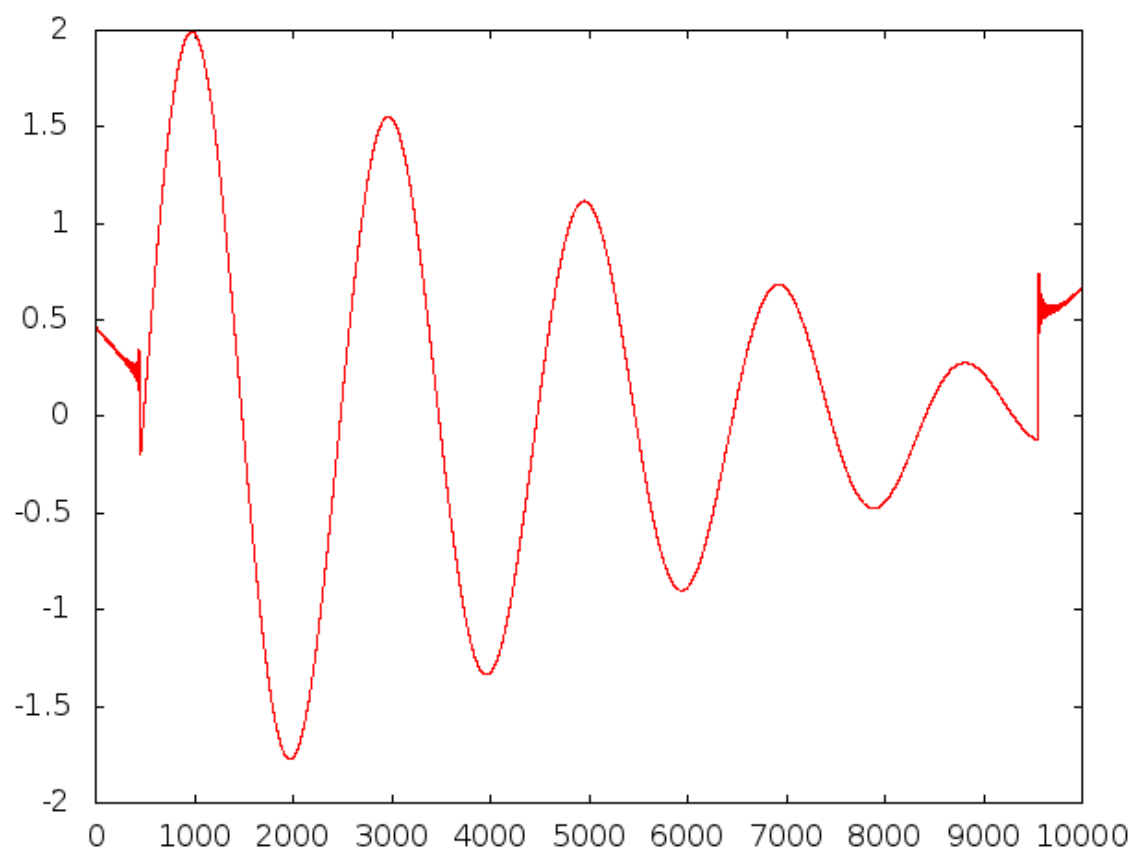


Figure 6: Cuda result of $i = 01000$.

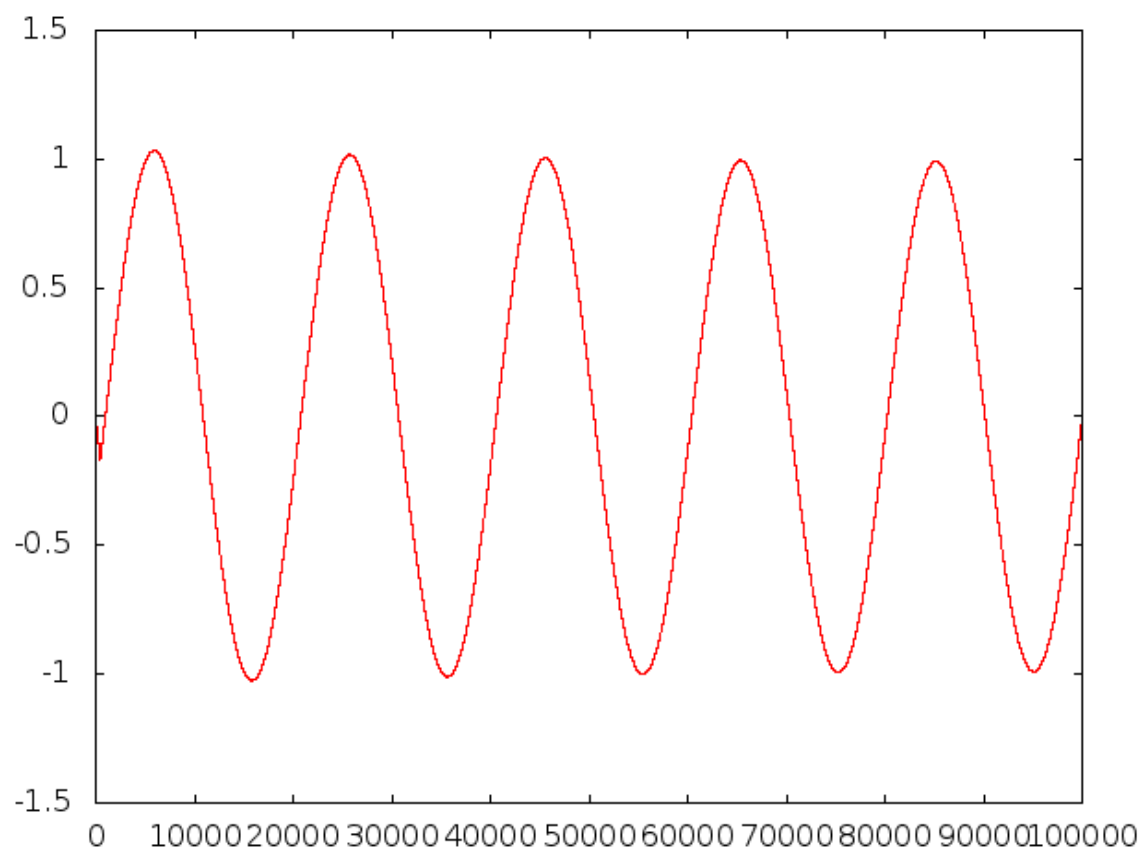


Figure 7: Correct result of $i = 100000$.

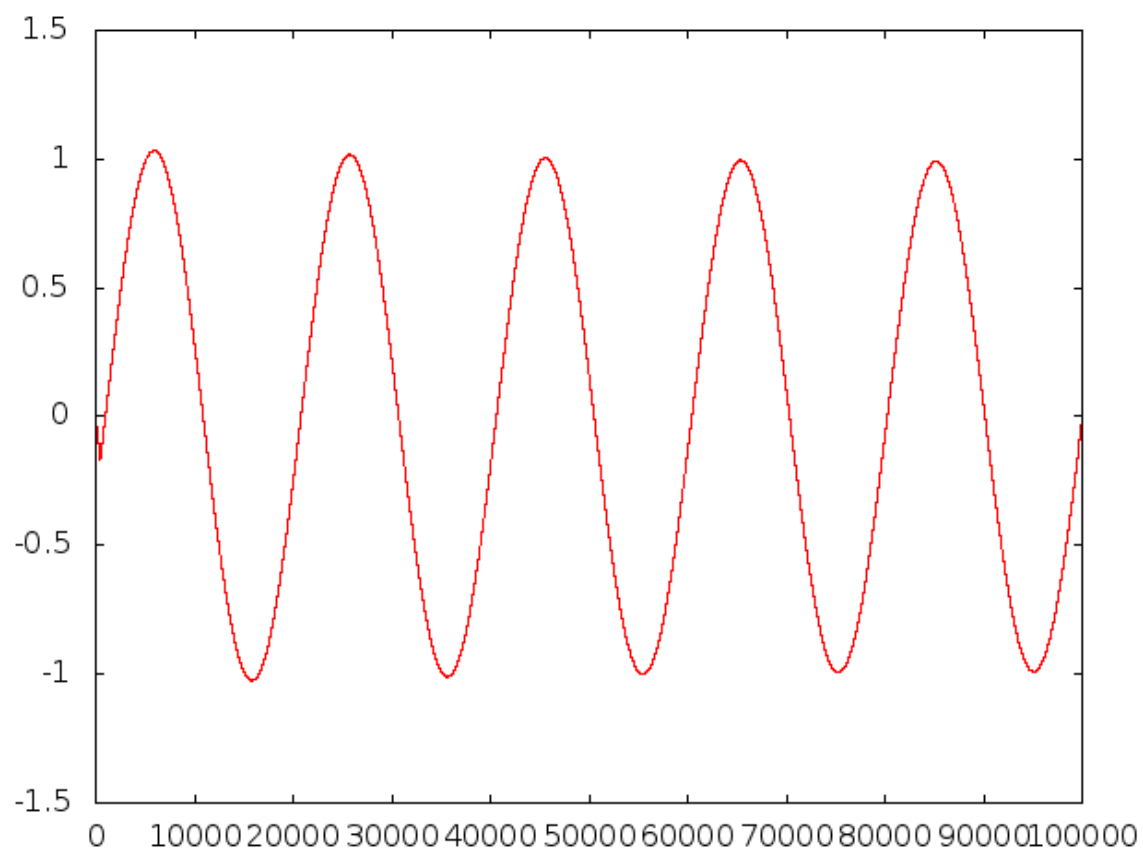


Figure 8: Cuda result of $i = 100000$.