

NSC miniproject 2

Kazim, Kocan
kkocan19@student.aau.dk

All the algorithms were run on a Ryzen 5 3600 at 4GHz with 32GB of RAM at 3000MHz. All implementations were run with a pixel density of 2000 making the array have a shape of (2000, 2000) and they were run for 500 iterations and all were locally executed.

The Mandelbrot algorithm has `numpy.all(numpy.abs(z) >= 2)` which makes it so that it shortens the execution time for the blocks located in regions where divergence is obtained quickly. This is true for all runs.

The execution time for numpy has been optimized by having the data types explicit, this is done for the all data types of clongcomplex (complex256), complex128, complex64, float128, float64, float32, and float16. The execution times for all numpy can be seen in figure 1

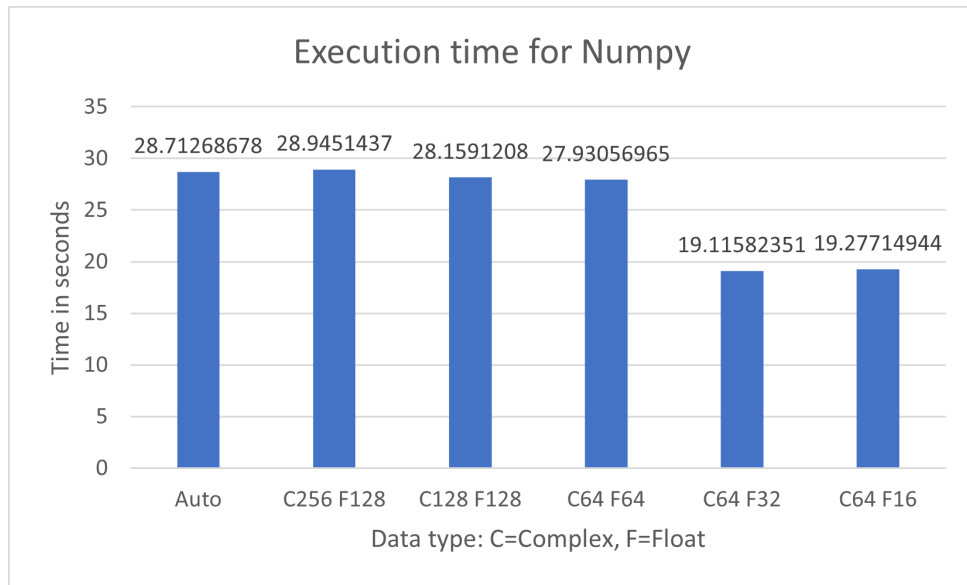


Figure 1

The execution time for all the different data types between dask and numpy can be seen in the following figures. Dask has 3 different chunk sized chosen from L1, L2, and L3 cache sizes from the CPU.

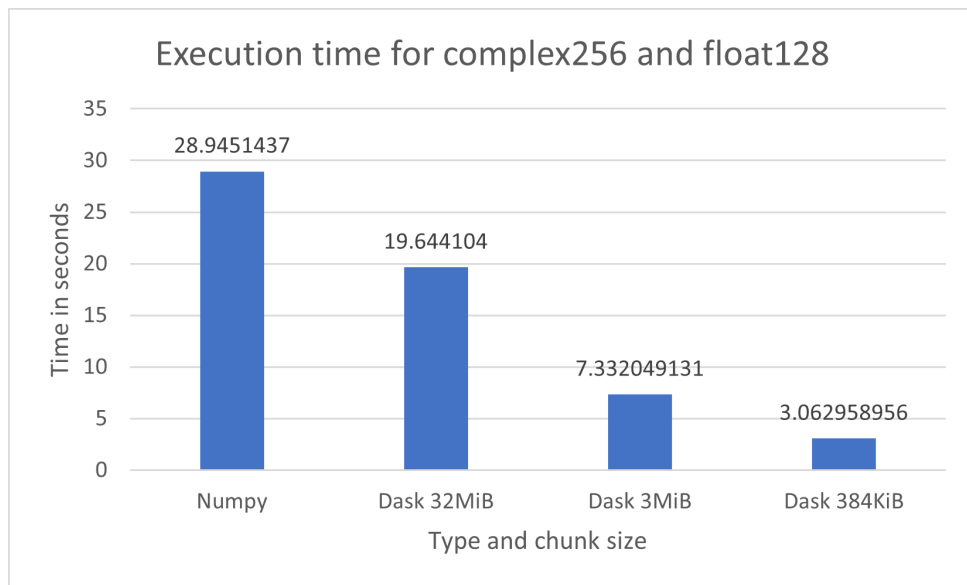


Figure 2

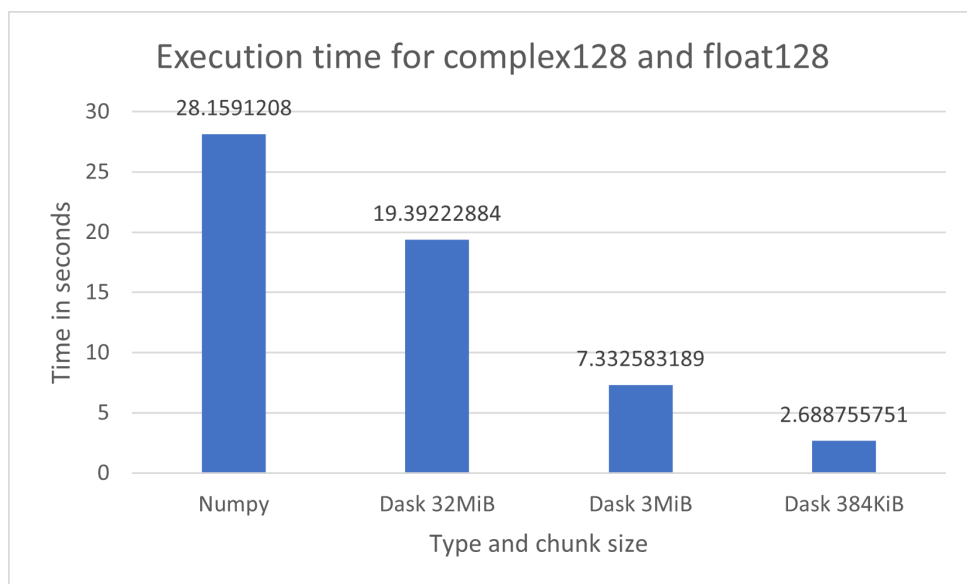


Figure 3

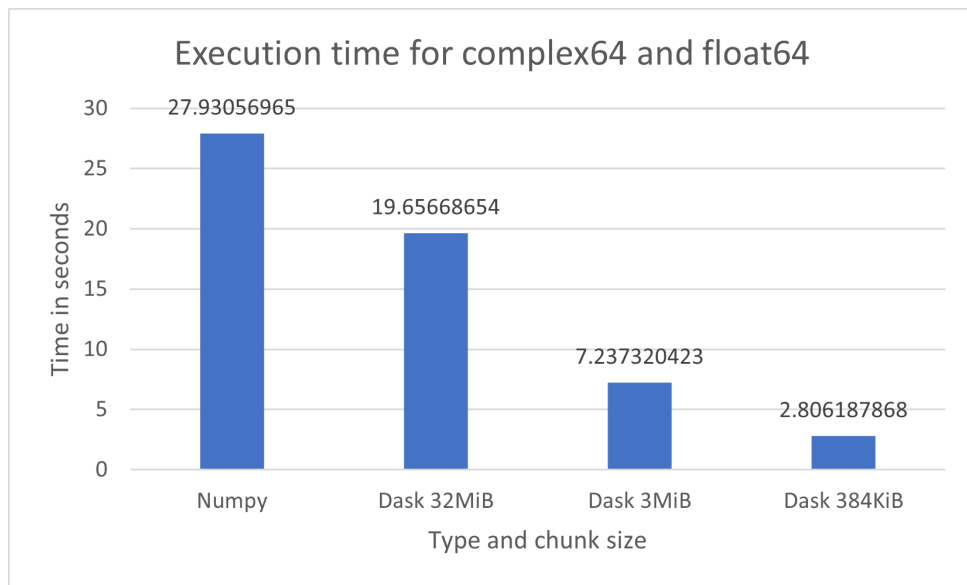


Figure 4

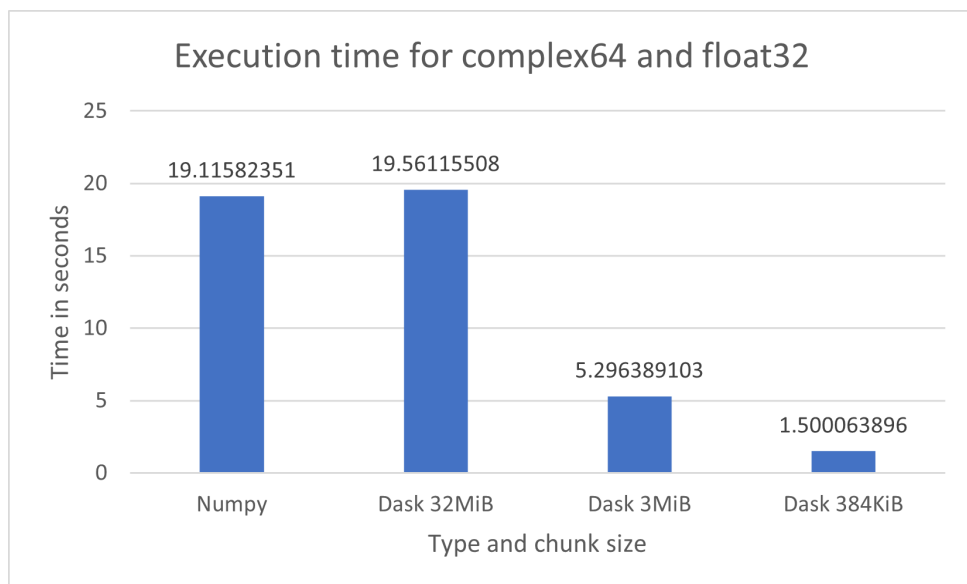


Figure 5

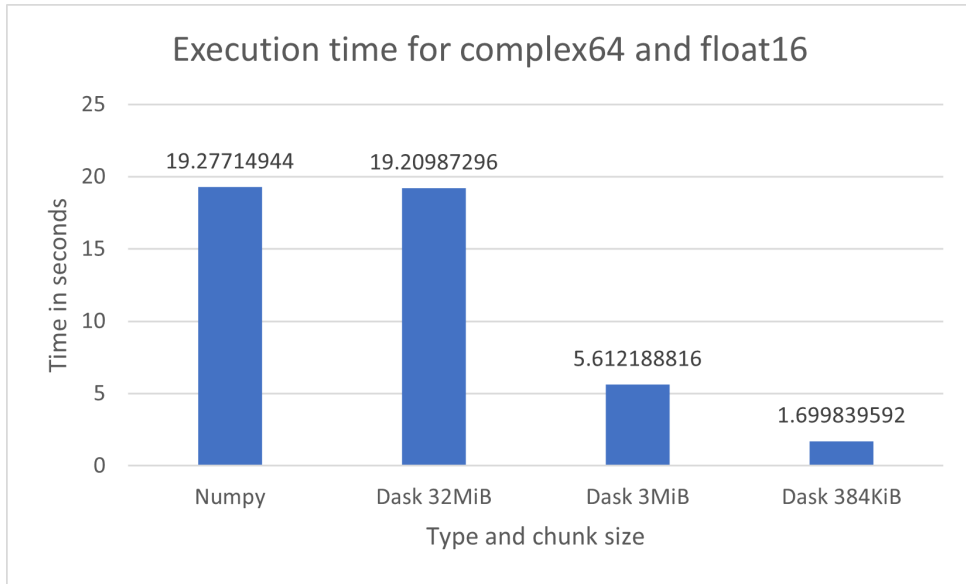


Figure 6

The dask implementation uses `da.from_array(input_arr, chunks=chunk_size)` which creates dask array data type from a numpy array. The difference between auto-assigned data type from numpy vs dask can be seen in figure 7

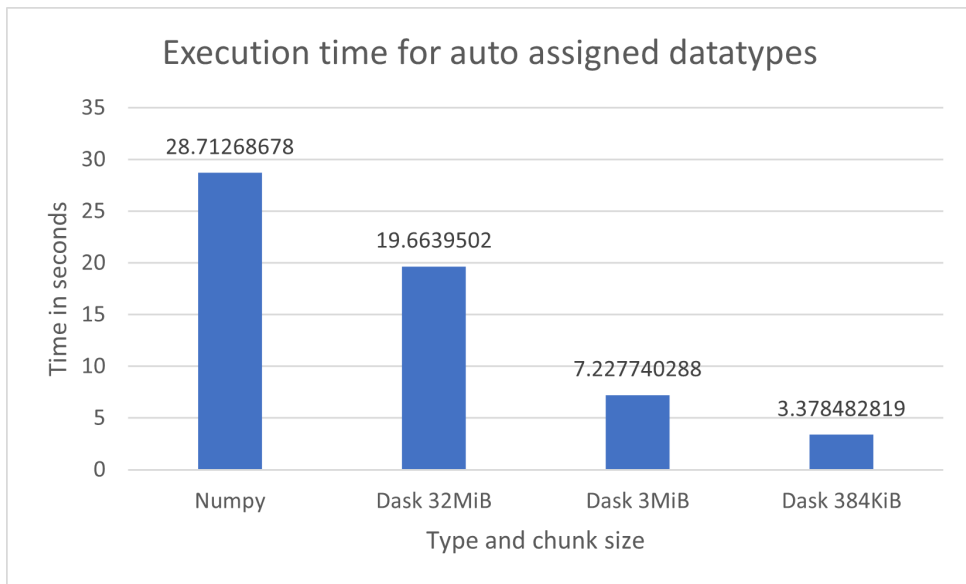


Figure 7

The difference between all dask implementations with 384KiB with different data type can be seen in figure 7

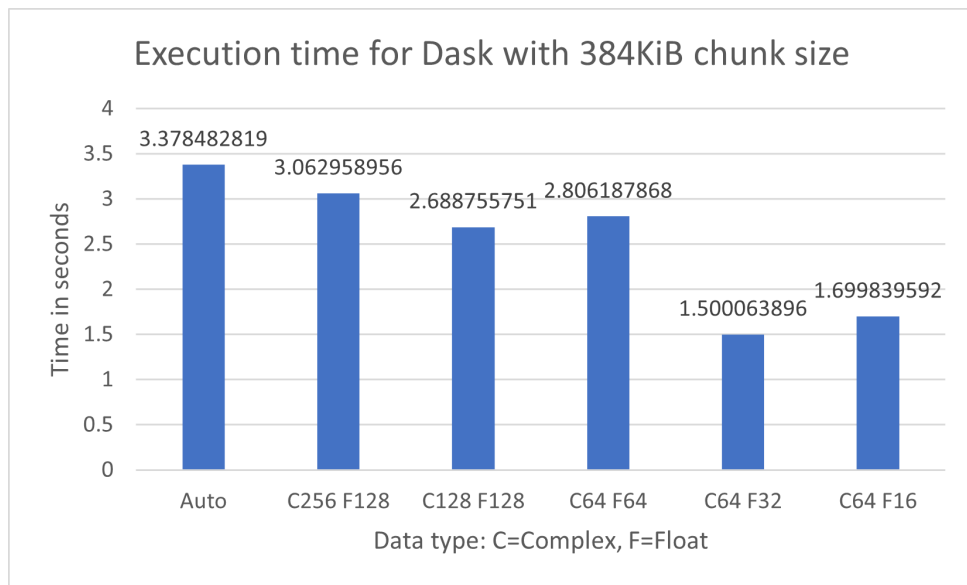


Figure 8