Final Project Phase 2

Members: Jonathan Jackson, Sriram Arjula, Ryan Luna

As an update to the complications mentioned in the phase 1 report, we are deciding to keep image processing a part of the project. The purpose being to expand on the applications of the research presented in the paper, as opposed to simply recreating the experiment in python. Moreover, looking into our code further, image processing is not the culprit of the slowdown in parallelized execution of the encryption algorithm. We are running into issues with parallelization in the python language itself. This can be observed in the following images. The left image displays the encryption/decryption time of the parallel algorithms, the right image displays that of the sequential implementation.

Text

Description automatically generated Text

Description automatically generated

The issues reside with python’s Global Interpreter Lock (GIL) which is essentially a mutex lock that protects access to python objects (and prevents race conditions). To overcome this, we need to work around using threads to achieve parallelism. This was currently being implemented using python’s threading package.

The referenced paper is using OpenCL to run computation on several Nvidia GPU cores. We speculate that it may be possible to work around the GIL issue using the pyopencl package, Python’s OpenCL package. Given the learning curve of implementing our code using pyopencl, pursuing this method does not seem feasible during the timeframe given. Another possible workaround is using the multiprocessor package. Instead of creating threads, this package enables the creation of several processes to run parallel code. However, in using the multiprocessor package, the creation of processes and inter-process communication induces additional overhead. We fear that this may leave us with the same result as before (with sequential executing faster than parallel).

These two solutions to our main issue are being analyzed aggressively by the team. On one end, using the OpenCL method ties our work closer to that of the project. This allows us to follow up on the initial objective of expanding the application scope of the research. On the other end, achieving successful parallelization within our timeframe seems more feasible using the multiprocessing package. However, this does not do any justice to expanding the scope of the research work of the paper being presented. At the time of writing, we are leaning towards the use of the multiprocessing package due to our timeframe and likelihood of deriving some solid results.