Concurrent & Parallel Systems - Coursework 2 Report

Glenn Wilkie-Sullivan - 40208762

November 23, 2018

Abstract

This report will comprehensively examine the approaches to parallelising a sequential JPEG compressor, taken from user kornelski on GitHub. These approaches will range from CPU-based parallelism to GPU frameworks and architectures such as CUDA and OpenCL. The ideal result is an optimised JPEG compressor which runs significantly faster with the new approaches implemented.

1 Introduction and Background

In order to understand how JPEG (Joint Photographic Experts Group) compression can be parallelised, we must first investigate the overall process and purpose of the compression. According to techradar.com, JPEG compression, "is a lossy compression format conceived explicitly for making photo files smaller and exploits the imperfect characteristics of our perception". The process for this can be split into five main steps:

- Covert RGB colours of the image to YCbCr (Luminance, Chroma: Blue; Chroma: Red) colour space.
- Preprocess image for DCT (Discrete Cosine Transformation) conversion.
- DCT transformation.
- Coefficient Quantization
- Lossless Encoding

We will touch more on these concepts in the following sections, which involve evaluating the base program, suggesting a better solution for optimisation, implementing it as described, discussing the results and wrapping up the findings. First, the program must be examined and evaluated based on it's overall performance.

2 Initial Analysis

For this project, the JPEG compressor created by the user 'kornelski' on GitHub will be analysed. The link for this repository can be found as a reference. The specifications used to run this compressor and analyse it are as follows:

- CPU: Intel Core i7-6700HQ @ 2.60GHz (4 cores, 8 threads)
- GPU: NVIDIA GeForce GTX 960M (4GB, GDDR5)

As such, the report will detail a methodology and results assuming these specifications. To start with, I used the Visual Studio 2017 diagnostic tools to analyse the overall CPU usage of the program when given various parameters. The program has a range of functionality, and the most pressing of them is the exhaustive test for the compressor, amongst the general compression and decompresion. First, we will look at the general compression algorithm, which has multiple variables affecting the execution time. Among these variables is the image size and quality factor. The quality factor is simply a number ranging from 0 - 100, relative to how sharp the resulting image should be. By modifying the quality factor and image size over 10 runs, the following results were found:

Image Size	Quality Factor: 40	Quality Factor: 60	Quality Factor: 80	Quality Factor: 100
100 x 100	0.005 seconds	0.006 seconds	0.010 seconds	0.010 seconds
200 x 200	0.0089 seconds	0.0213 seconds	0.0129 seconds	0.0177 seconds
400 x 400	0.0337 seconds	0.0339 seconds	0.0369 seconds	0.0445 seconds
800 x 800	0.1309 seconds	0.1343 seconds	0.1492 seconds	0.1797 seconds
1600 x 1600	0.3461 seconds	0.3586 seconds	0.381 seconds	0.4653 seconds
3000 x 3000	1.0833 seconds	1.0854 seconds	1.1238 seconds	1.2284 seconds

Table 1: JPEG Compression Execution Time

These findings were exlusively within Visual Studio, Release mode, x86. As we can see from table 1, the program itself is very fast. The quality factor of the compression doesn't seem to have much effect on the execution time - in most cases, there was only a rise of roughly 3.5%. This table will eventually be used as a comparison after a parallelised solution is implemented - for now, we have to identify the bottleneck(s) of the program, so that we have a foundational understanding of where the program can be parallelised.

3 Methodology

4 Results and Discussion

5 Conclusion

References

[Dias,] Dias, D. Jpeg compression algorithm. https://medium.com/@danojadias/jpeg-compression-algorithm-969af03773da.

[kornelski,] kornelski. jpeg-compressor. https://github.com/kornelski/jpeg-compressor.

[Plus,] Plus, P. All you need to know about jpeg compression. https://www.techradar.com/news/computing/all-you-need-to-know-about-jpeg-compression-586268/2.