

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
- 3 Methodology
- 4 Results and Discussion
- 5 Conclusion

References

[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>.