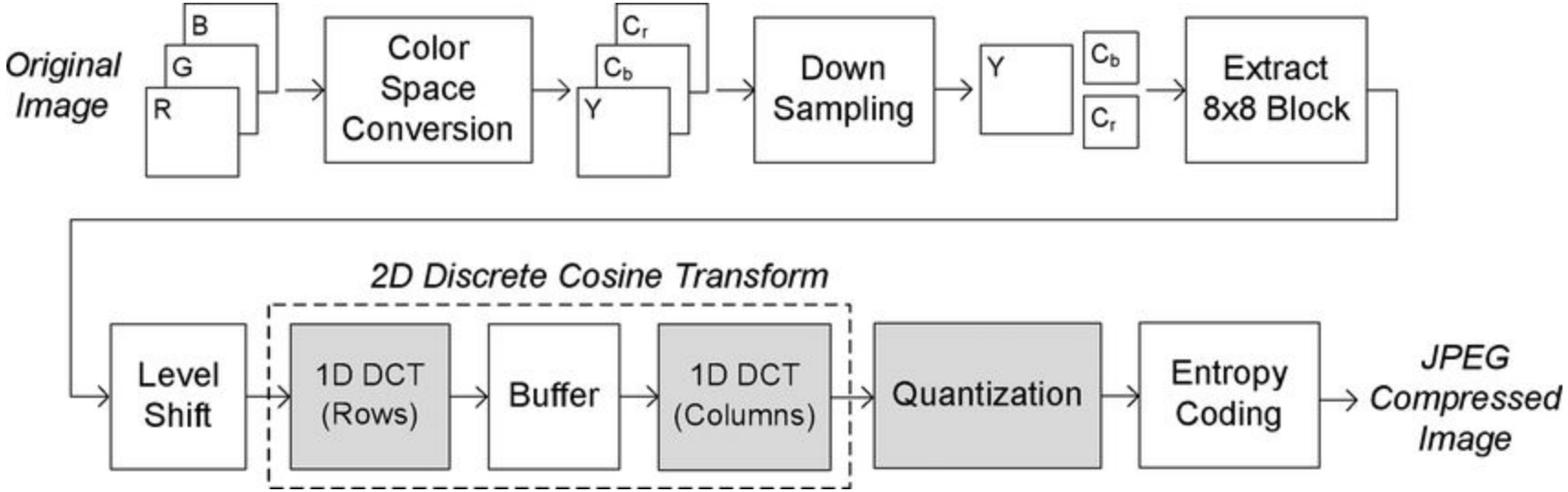
Parallel JPEG Compression Algorithm

Rama Mannava, Spencer Yu

Summary: We explored the performance of the JPEG algorithm when parallelized using OpenMPI and OpenMP across different workloads. Image compression is a difficult problem to parallelize because it involves many data dependencies as it is a pipeline of operations, and involves computation on large amounts of data (typically millions of pixels across multiple color channels). We found the OpenMPI implementation to work well for smaller images but scale poorly to larger images, while the OpenMP workload performed well across both large and small images.

Algorithm


Original Image
B
G
R
Color Space Conversion
Cr
Cb
Y
Down Sampling
Y
Cb
Cr
Extract 8x8 Block
2D Discrete Cosine Transform
Level Shift
1D DCT (Rows)
Buffer
1D DCT (Columns)
Quantization
Entropy Coding
JPEG Compressed Image

Parallelization

Environment: ghc cluster computers

- Latedays not necessary due to use case (JPEG)

OpenMPI: message passing (on CPU)

- (De)serialization: had to create mirror structures that did not use pointers
- High overhead: from MPI operations on encoded blocks
- Performed 1.44-1.66x better than seq for small images (~500x500)
- Performed far worse than seq for large images (~4000x2000)

OpenMP: shared memory (on CPU)

- Low overhead: no need to serialize output repeatedly
- Performed 2.0-2.5x better than seq across both large and small images

CUDA: massively parallel (on GPU)

- Decided to not use this parallelization approach
- Speculation: less efficient due to excessive copying overhead

Parallelization: OpenMPI

Message Passing Locations: Split at natural divergence in task type

1. Scatter: Give each thread a section of bytes to convert into image
2. Gather: YCbCr resampled sections of image to reduce into full image
3. Scatter: Give each thread blocks to compress
4. Gather: Encoded blocks that (decoded and combined) form the full image

(De)serialization: major source of overhead and barrier to scalability

Mirror copies of data structures for buffer (gather encoded blocks)

Need structures that do not use pointers

Lack of pointers contradicts with std::map required for encoding

Parallelization: OpenMP

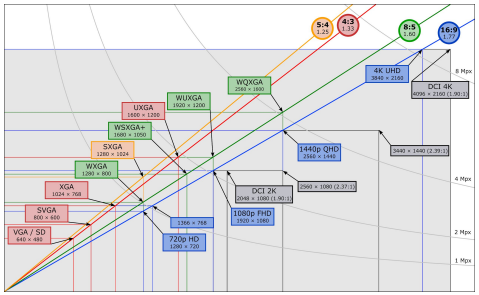
Pragma Locations: Loops over macroblocks

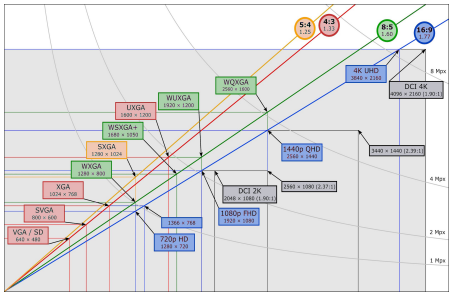
- Tasks get a subset of image blocks for each step of the pipeline
- Avoid useless overhead: loops chosen only if sufficient computation done

Pragma Decisions: #pragma omp parallel for

- Relatively uniform loads across each macroblock and task instance
- Outperformed competing methods (guided scheduling, dynamic scheduling)
- Thread computation times are uniform
- No recursion: task limit not required

Testing Images





Results

Performance: cameraman.png (495x500)

Step	Sequential	MPI	OMP
Load Image	0.020s	0.020s	0.019s
Setup MPI	N/A	0.000s	N/A
Convert Bytes to Image	0.047s	0.008s	0.032s
Convert RGB to YCbCr	0.044s	0.007s	0.027s
Gather YCbCr Pixels	N/A	0.046s	N/A
Convert YCbCr to Blocks	0.047s	0.094s	0.043s
DCT	0.007s	0.002s	0.004s
Quantize/DPCM	0.016s	0.002s	0.010s
RLE	0.208s	0.028s	0.047s
Gather Encoded Blocks	N/A	0.063s	N/A
Encode Compressed Image	0.002s	0.002s	0.002s
Total Time	0.394s	0.273s	0.184s
Speedup	**1x**	**1.44x**	**2.14x**

Performance: cookie.png (2796x1414)

Step	Sequential	MPI	OMP
Load Image	0.198s	0.166s	0.181s
Setup MPI	N/A	0.000s	N/A
Convert Bytes to Image	0.411s	0.075s	0.306s
Convert RGB to YCbCr	0.459s	0.066s	0.269s
Gather YCbCr Pixels	N/A	0.475s	N/A
Convert YCbCr to Blocks	0.534s	1.220s	0.485s
DCT	0.093s	0.029s	0.064s
Quantize/DPCM	0.211s	0.044s	0.154s
RLE	2.970s	0.432s	0.665s
Gather Encoded Blocks	N/A	9.283s	N/A
Encode Compressed Image	0.005s	0.010s	0.005s
Total Time	4.873s	11.801s	2.131s
Speedup	**1x**	**0.41x**	**2.29x**

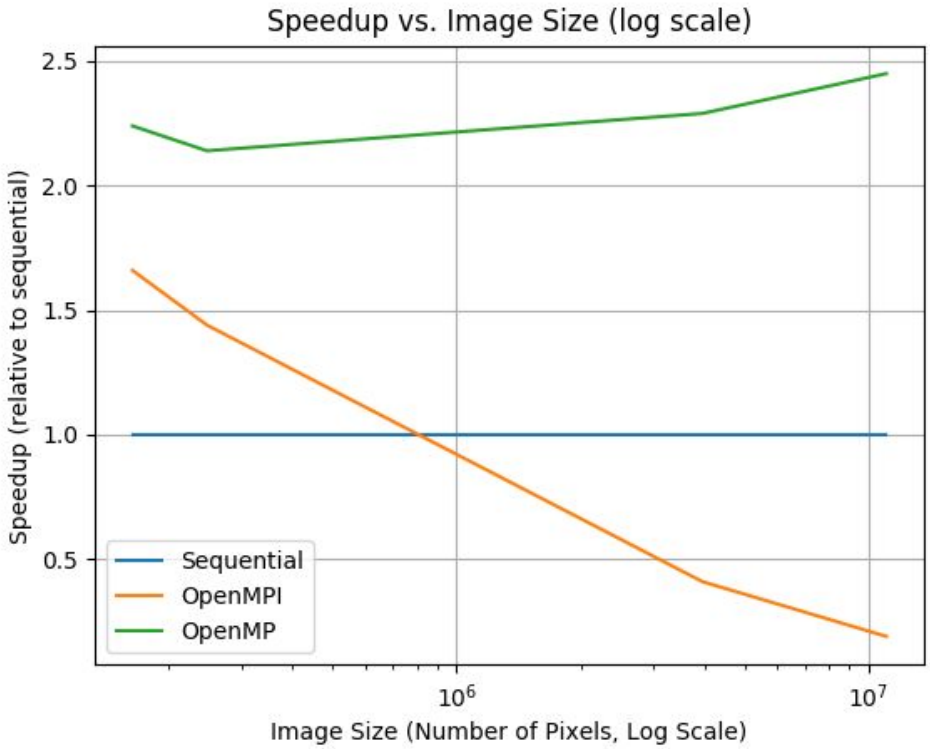
Performance: peppers.png (508x381)

Step	Sequential	MPI	OMP
Load Image	0.023s	0.018s	0.020s
Setup MPI	N/A	0.000s	N/A
Convert Bytes to Image	0.041s	0.006s	0.023s
Convert RGB to YCbCr	0.038s	0.005s	0.020s
Gather YCbCr Pixels	N/A	0.030s	N/A
Convert YCbCr to Blocks	0.040s	0.075s	0.032s
DCT	0.006s	0.001s	0.003s
Quantize/DPCM	0.014s	0.002s	0.011s
RLE	0.169s	0.023s	0.040s
Gather Encoded Blocks	N/A	0.042s	N/A
Encode Compressed Image	0.003s	0.001s	0.001s
Total Time	0.336s	0.202s	0.150s
Speedup	**1x**	**1.66x**	**2.24x**

Performance: reschart.png (4320x2560)

Step	Sequential	MPI	OMP
Load Image	0.291s	0.261s	0.197s
Setup MPI	N/A	0.000s	N/A
Convert Bytes to Image	1.162s	0.236s	0.771s
Convert RGB to YCbCr	1.242s	0.226s	0.742s
Gather YCbCr Pixels	N/A	1.243s	N/A
Convert YCbCr to Blocks	1.516s	3.431s	1.262s
DCT	0.269s	0.085s	0.151s
Quantize/DPCM	0.605s	0.126s	0.395s
RLE	7.508s	1.099s	1.597s
Gather Encoded Blocks	N/A	59.956s	N/A
Encode Compressed Image	0.011s	0.010s	0.021s
Total Time	12.612s	66.674s	5.147s
Speedup	**1x**	**0.19x**	**2.45x**

Results


Speedup vs. Image Size (log scale)
Speedup (relative to sequential)
2.5
2.0
1.5
1.0
0.5
Sequential
OpenMPI
OpenMP
10^6
10^7
Image Size (Number of Pixels, Log Scale)