

# Simple Huffman compression on a GPU

Emanuel Poremba



- Compression is used in many tasks.
- Limited often by CPU power.
- Huffman is easy to understand.
- Lossless compression.
- Most parts can be parallelized.

# Huffman codec – HowTo (Serial approach)

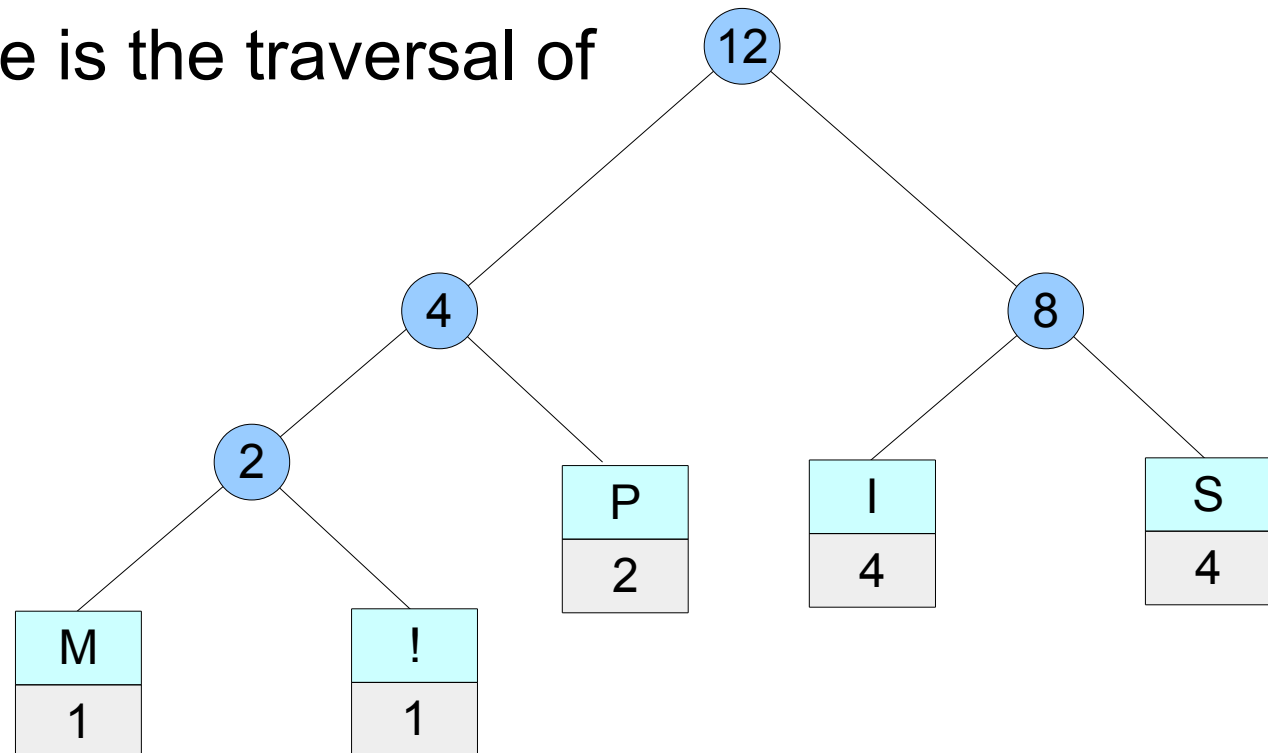
- Create a histogram of all used symbols

MISSISSIPPI! →

M	I	S	P	!
1	4	4	2	1

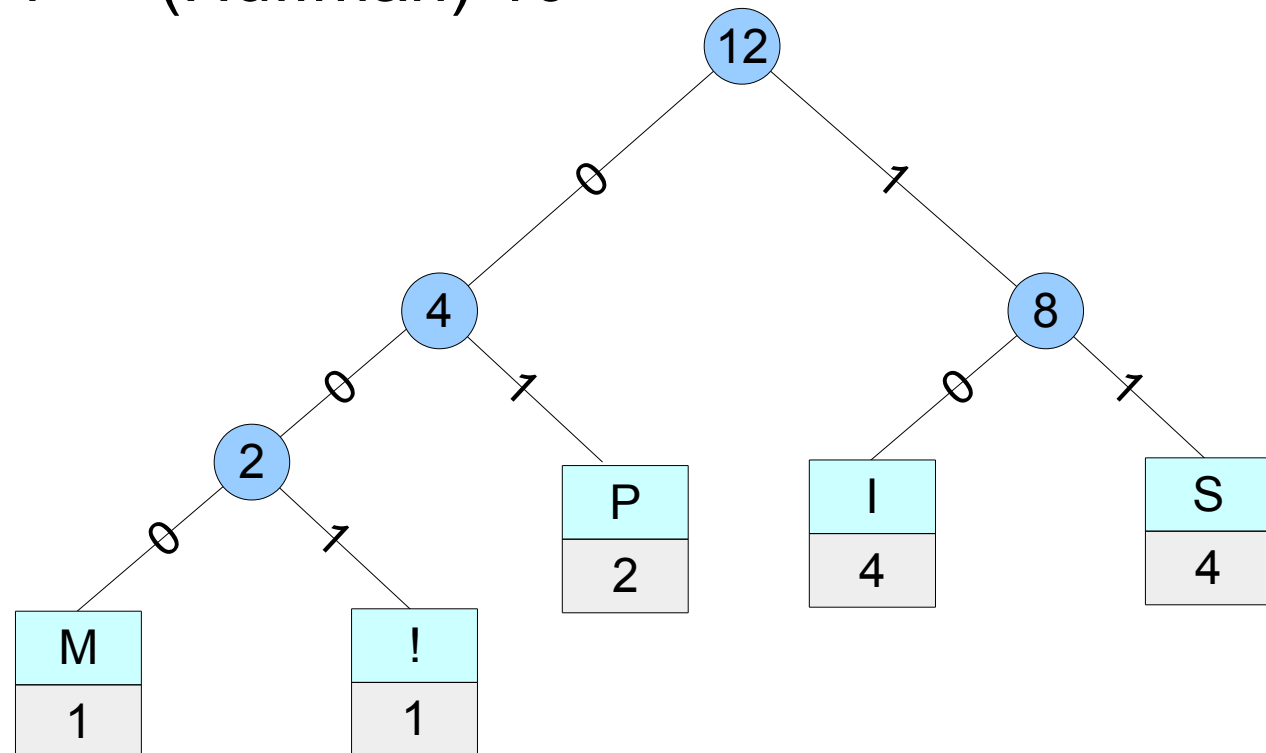
# Huffman codec – HowTo (Serial approach)

- Create an unbalanced tree
  - Level of symbol in tree depends on propability of symbol
  - New symbol code is the traversal of the tree.



# Huffman codec – HowTo (Serial approach)

- Symbol compression:
  - M (Ascii) 1001101 → (Huffman) 000
  - I (Ascii) 1001101 → (Huffman) 10
  - ...

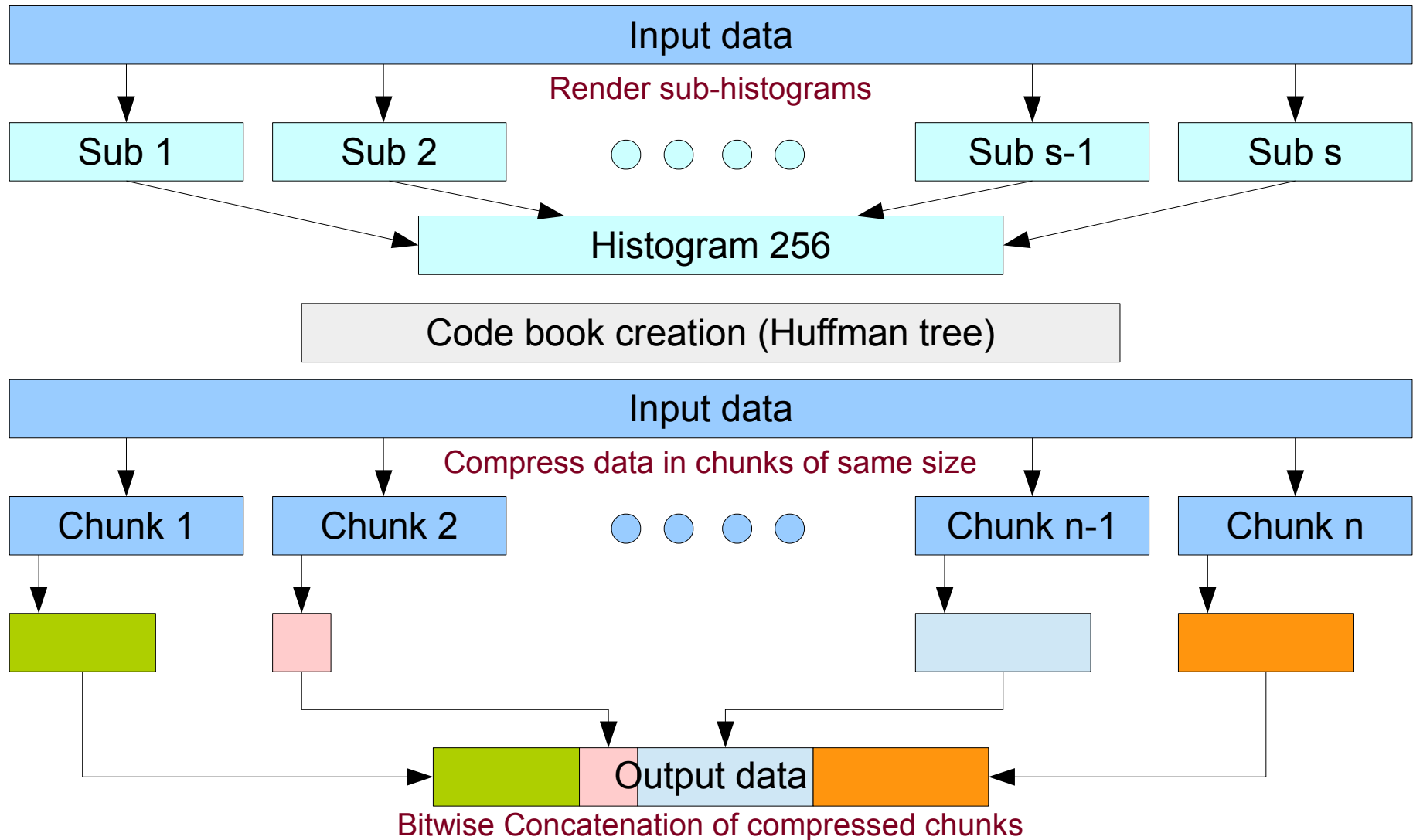


# Huffman codec – Overview

- First step: Histogram
- Second step: Creation of Huffman tree / Code book
- Third step: Compression

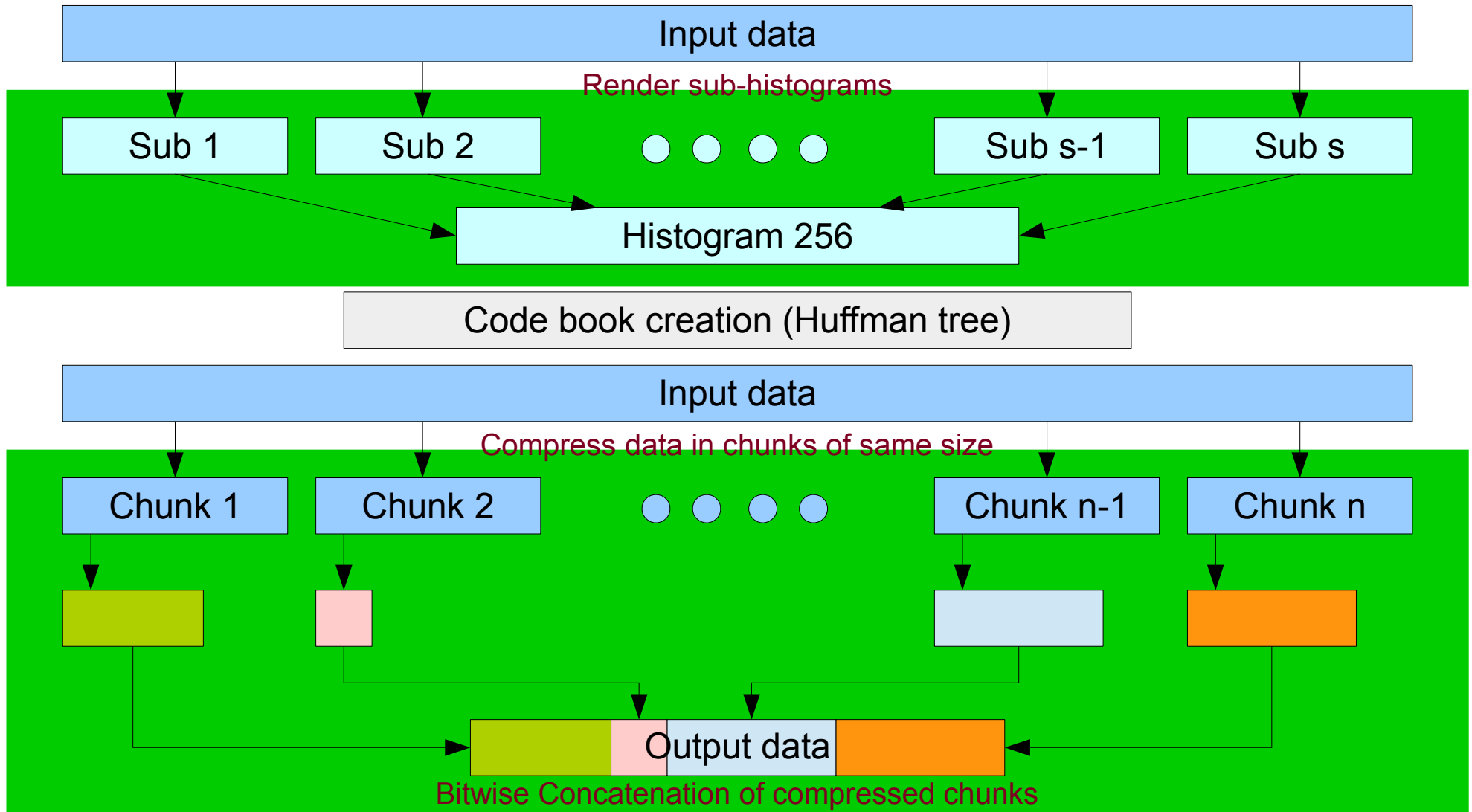
- First step: Histogram
  - Chunks of input data create sub-histograms in parallel
  - Merge these sub-histograms to one
- Second step: Creation of Huffman tree / Code book
  - Stays the same because this does need nearly no time
- Third step: Compression
  - Compress chunks of input data
  - Calculate the compression chunk offsets (PPS)
  - Concatenate the chunks to one compressed data block

# Huffman codec – Compression chunks



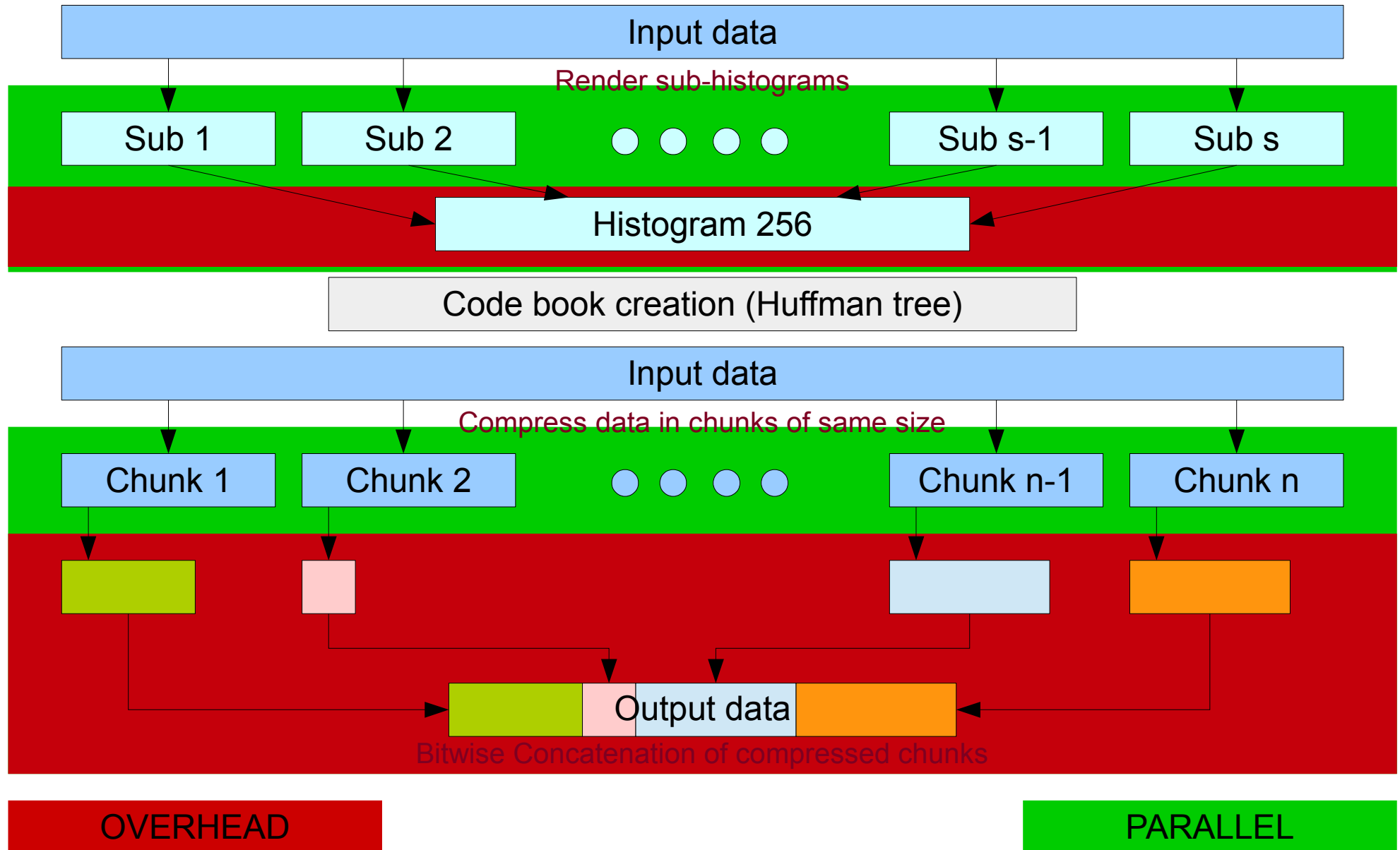


# Huffman codec – Compression chunks

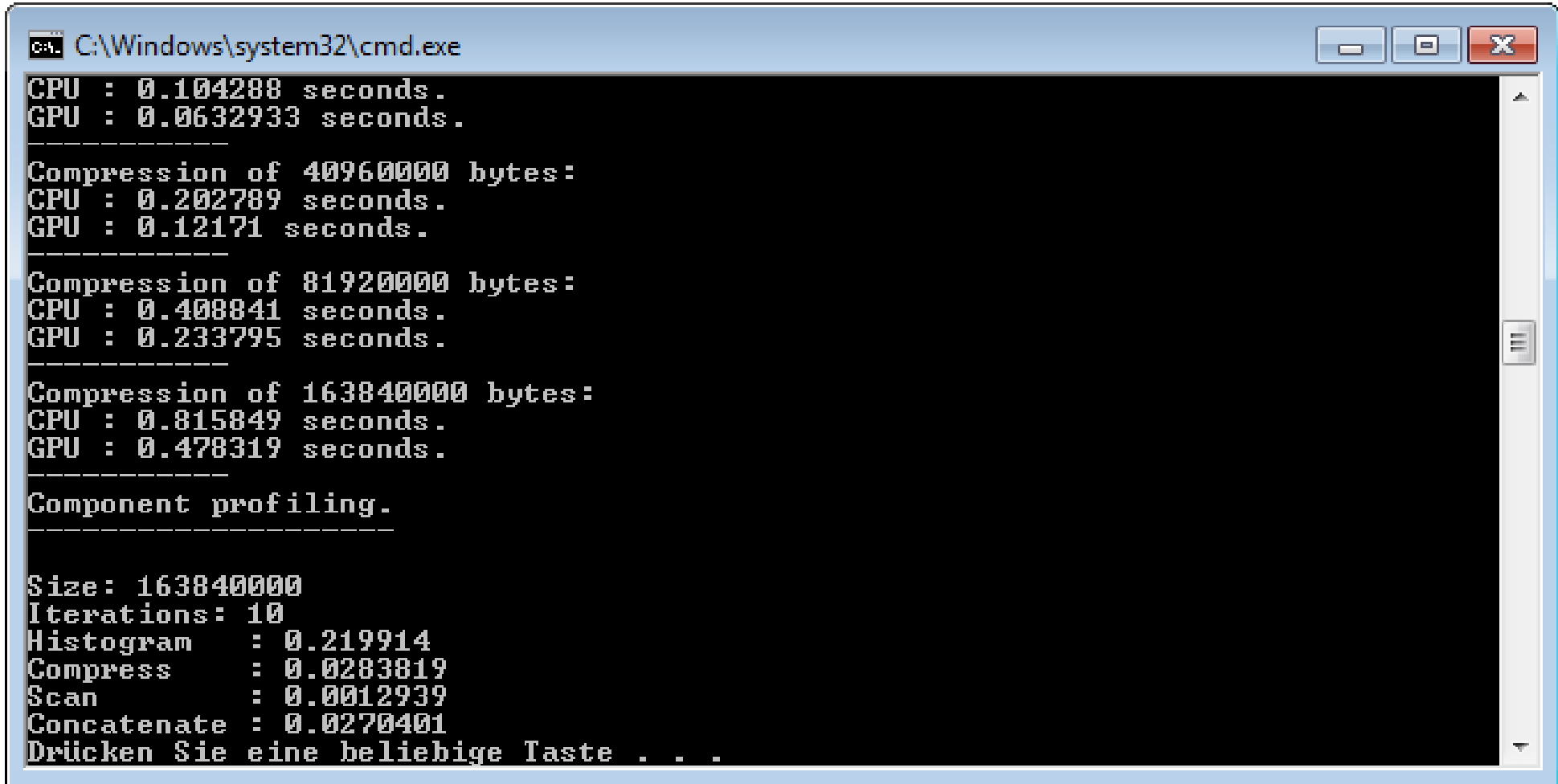


PARALLEL

# Huffman codec – Compression chunks

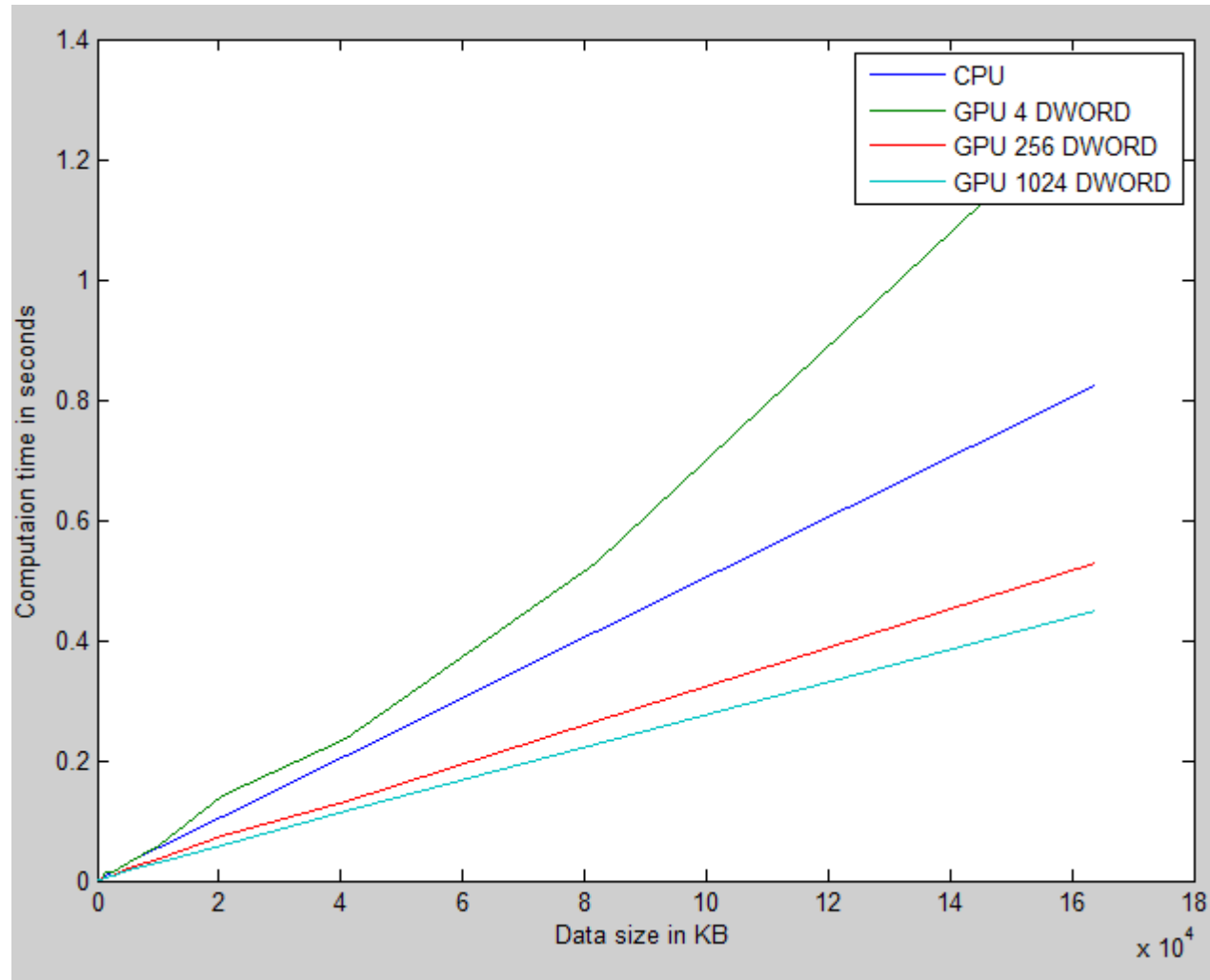


# Huffman codec – Screen output

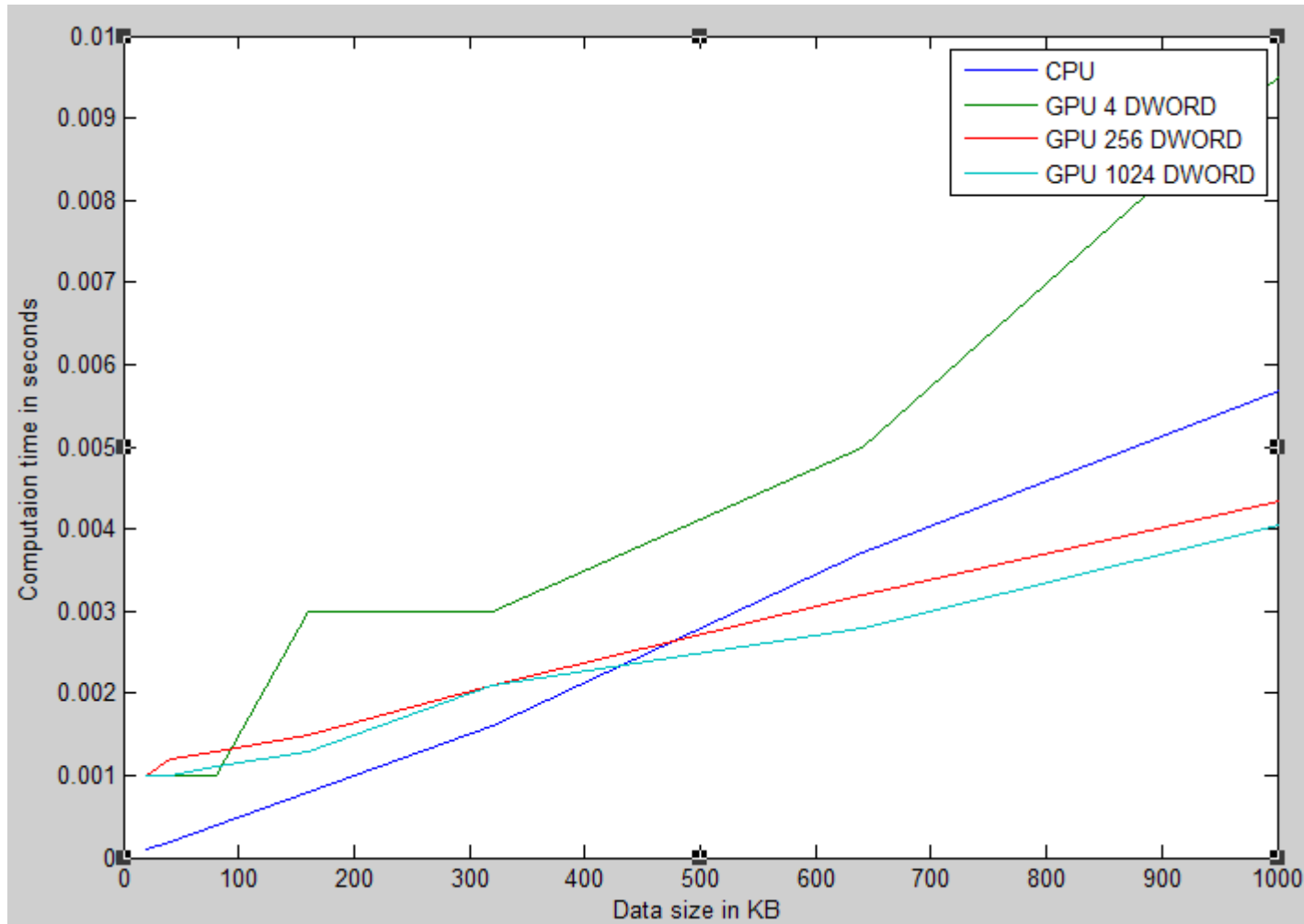


```
C:\Windows\system32\cmd.exe
CPU : 0.104288 seconds.
GPU : 0.0632933 seconds.
-----
Compression of 40960000 bytes:
CPU : 0.202789 seconds.
GPU : 0.12171 seconds.
-----
Compression of 81920000 bytes:
CPU : 0.408841 seconds.
GPU : 0.233795 seconds.
-----
Compression of 163840000 bytes:
CPU : 0.815849 seconds.
GPU : 0.478319 seconds.
-----
Component profiling.
-----
Size: 163840000
Iterations: 10
Histogram   : 0.219914
Compress    : 0.0283819
Scan        : 0.0012939
Concatenate : 0.0270401
Drücken Sie eine beliebige Taste . . .
```

# Huffman codec – Big data sets

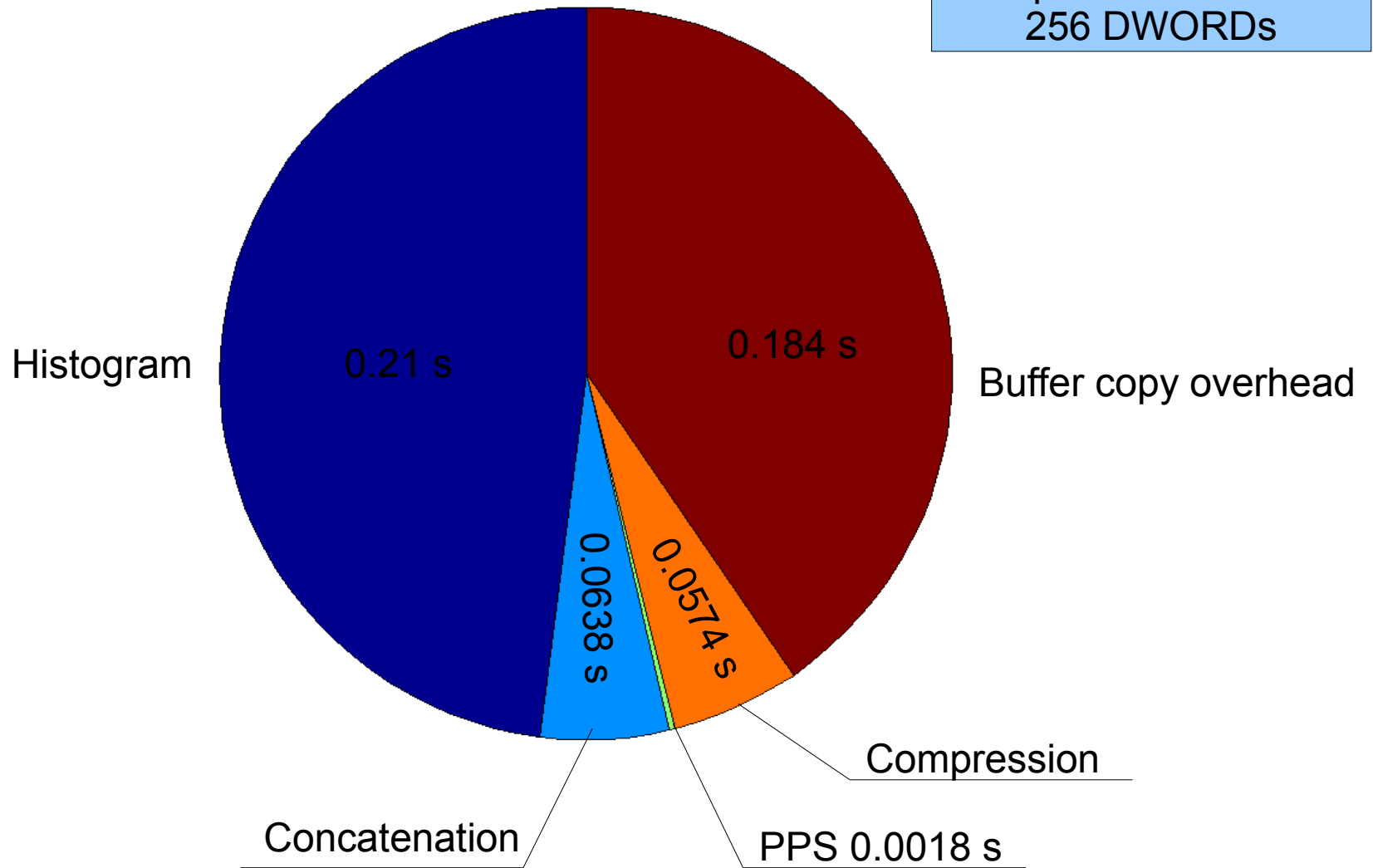


# Huffman codec – Small data sets



# Huffman codec – Algorithm analysis

Compression of 16 MB  
256 DWORDS



# Huffman codec – Summary

- GPU is faster on data sets greater than approx. 500 KB
- BUT: CPU implementation is not optimized.
- Compared to open source compression applications:  
GPU slower
- Big overhead of parallelization
- Pro: CPU can be used for other tasks while compressing on the GPU.
  - e.g.: File 1 compression on CPU, File 2 compression on GPU, and so on..