Universität Heidelberg
Institute for Computer Engineering (ZITI)

Master of Science Computer Engineering

GPU Computing

Exercise 6

Group gpu04
Pingitzer, Danny
Altuntop, Ekrem
Junge, Andreas G.

Due date Wednesday, December 11th, 09:00

6 Exercise

6.1 Reading

Read the following paper and provide review as explained in the first lecture (see slides):

Samuel Williams, Andrew Waterman, and David Patterson. 2009. Roofline: an insightful visual performance model for multicore architectures. Commun. ACM 52, 4 (April 2009), 65-76.

The article ”Roofline: an insightful visual performance model for multicore architectures.” by Samuel Williams, Andrew Waterman, and David Patterson is presenting a performance model, ”Roofline”, to estimate performance of a kernel on a certain system (multicore, accelerators). The proposed Roofline model ties together floating-point performance, operational intensity, and memory performance in a 2D graph. Furthermore it can show inherent hardware limitations for a given kernel and unveil benefit of optimizations.
It relies on the concept of Operational Intensity, this is the ratio of total floating-point operations to total data movement (bytes).
The performance in the model is bound by the peak flop rate and the streaming bandwidth, so basically the memory and compute boundness of a system. At the ridge point, the intersection of the diagonal (bandwith) and the horizontal (peak flop) roof, the x-coordinate is the minimum operational intensity required to achieve maximum performance and also a hint to the level of difficulty to achieve peak performance.
Since the model also takes regard to the memory boundness, we can also estimate, how hardware changes will affect the performance of our kernels.

I myself am not an expert in performance evaluation models, but the proposed model seems reasonable. It fits in the current computing era better than many older models, and is widely accepted in the HPC community. Still, older models, such as Amdahl’s Law are still relevant.

Review: Accept.

6.2 Reduction – CPU sequential version

We performed 100 iterations for robust measurement.

Figure 1: CPU Version

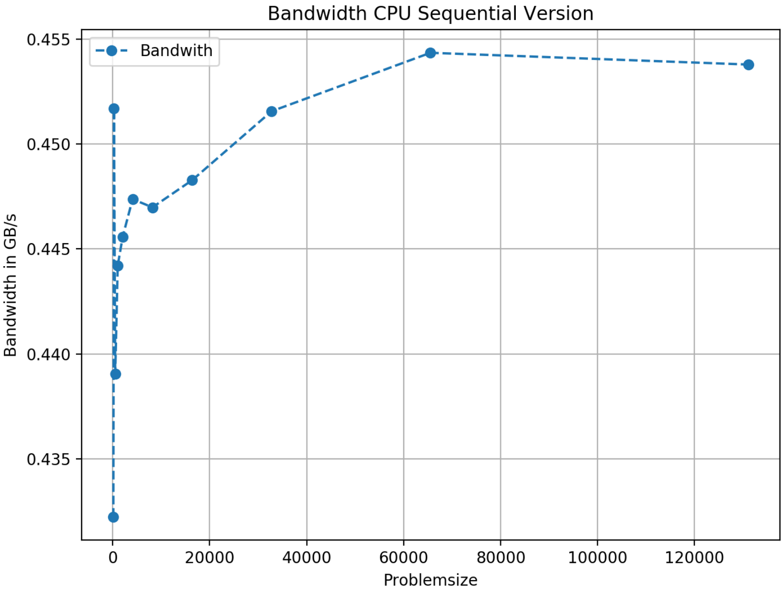
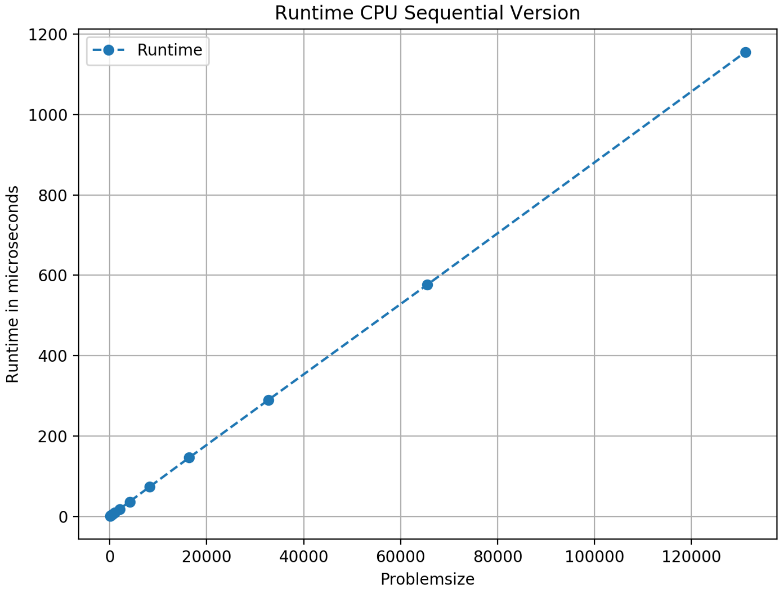


Figure 2: CPU Version



6.3 Reduction – GPU parallel initial version

We performed 200 iterations for robust measurement.

Figure 3: GPU Unoptimized Version

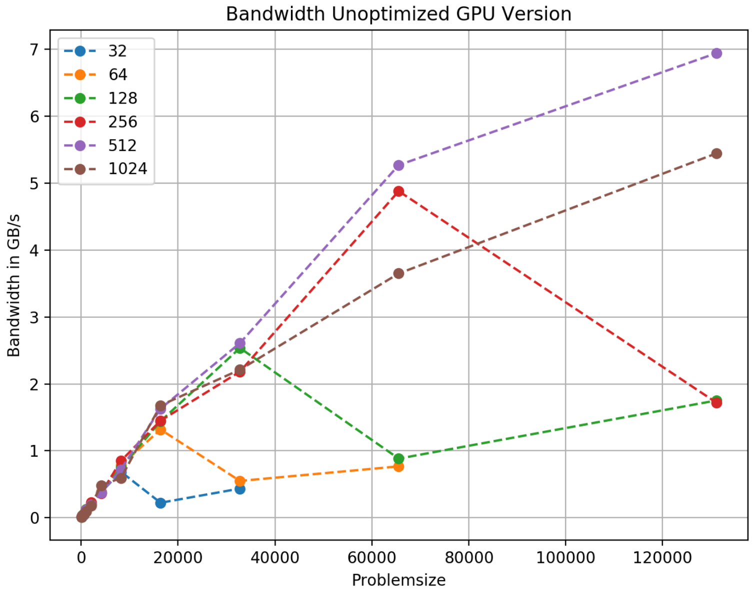
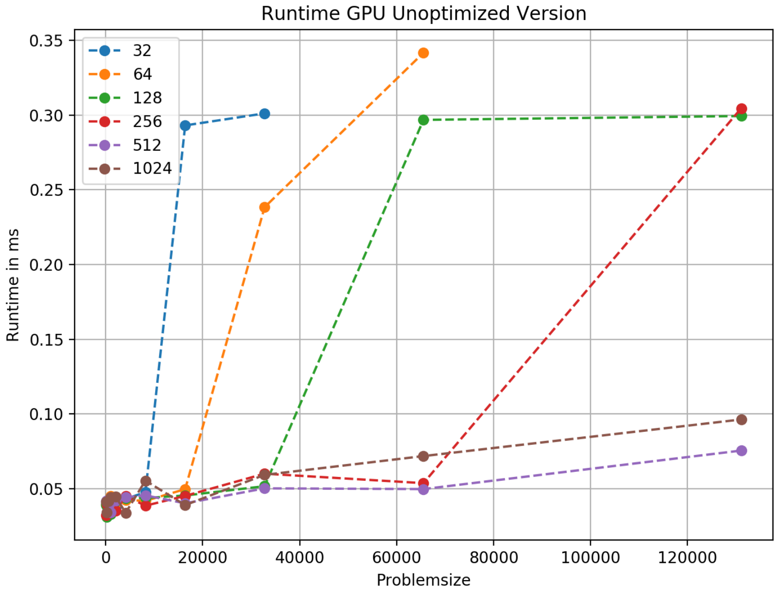


Figure 4: GPU Unoptimized Version



6.4 Reduction – GPU parallel optimized version

We performed 200 iterations for robust measurement.

Figure 5: GPU Optimized Version

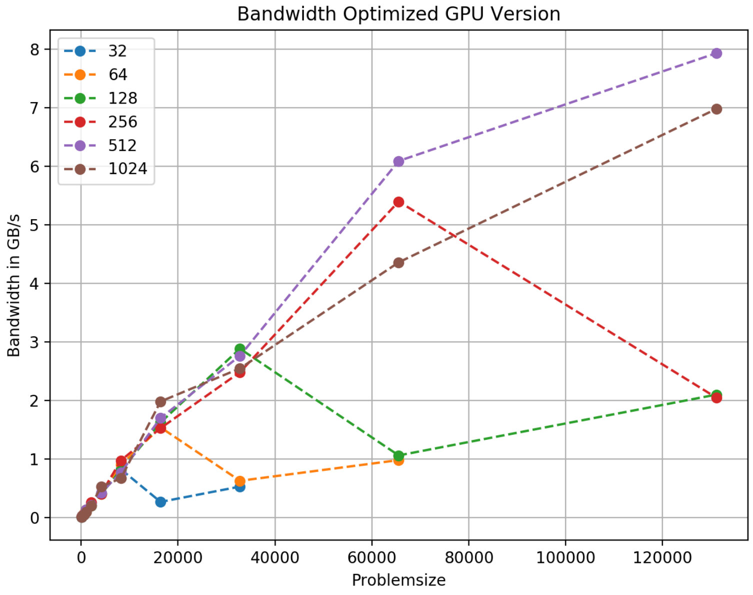
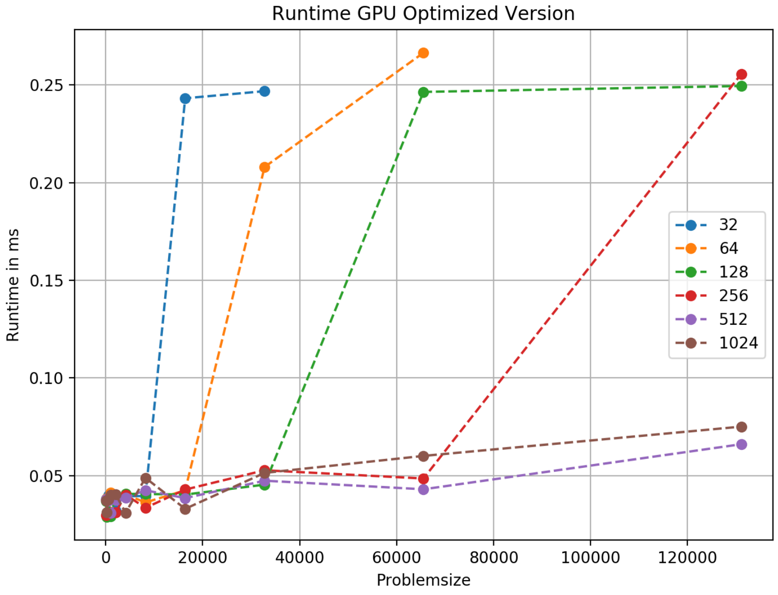
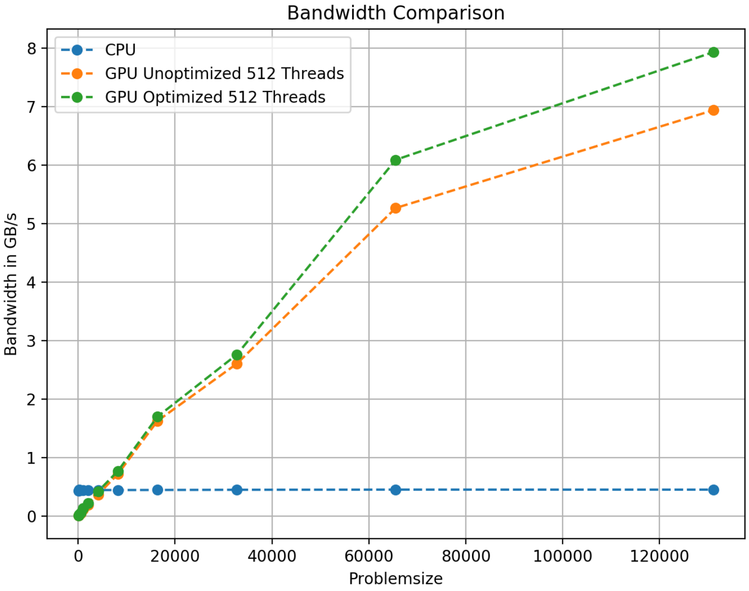


Figure 6: GPU Optimized Version



Compare sustained bandwidth over CPU and initial GPU versions.:
512 Threads in both versions showed the best performance in our tests.

Figure 7: Comparison


Bandwidth Comparison
CPU
GPU Unoptimized 512 Threads
GPU Optimized 512 Threads
Bandwidth in GB/s
Problemsize