LLPP: Assignment 4

Arveen Emdad Hamit Efe Çınar Iason Kaxiras

8 March 2024

1 Documentation

To run the program, simply invoke it like so:

```
$ demo/demo scenario.xml
```

The program will use CUDA to compute the heatmap by default and must be recompiled to run sequentially.

2 Machine Specifications

The experiments were ran on the Snowy compute cluster with the following specifications:

”Snowy consists of 228 compute servers (nodes) where each compute server consists of two 8-core Xeon E5-2660 processors running at 2.2 GHz. We provide 198 nodes with 128 GB memory (s1-s120, s151-s228) , 13 nodes with 256 GB (s138-s150) and 17 nodes with 512 GB (s121-s137). All nodes are interconnected with a 2:1 oversubscribed FDR (40 GB/s) Infiniband fabric. In total Snowy provides 3548 CPU cores in compute nodes” [1].

3 Questions

A. The memory accesses within a warp are ideally coalesced. Coalesced memory access refers to accessing memory locations in a contiguous and sequential manner by threads within a warp. This allows for efficient memory transactions and minimizes memory access latency.

When threads within a warp access consecutive memory locations, the memory requests can be coalesced into a single memory transaction. This means that instead of multiple individual memory transactions, the hardware can efficiently fetch a single block of memory containing data for all the threads in the warp. This improves memory bandwidth utilization and overall performance.

B. See Figure 2 for a plot comparing the execution times of the individual heatmap computation steps. Due to the heatmap creation steps having a large amount of atomic operations, that can explain why that part of the heatmap computation takes the greatest amount of time compared to scaling and blurring.

C. We observe about a speedup of around 18x compared with the sequential version computed on the CPU. There are many reasons as to why we do not achieve an N times speedup. First, there are many parts in the heatmap calculation that must be done synchronously, such as loading the desired positions from the agents into the GPU. Similarly, in the blur kernel the threads must reach a barrier before they are able to start computation of the tiles. It could also be that some parts of the computation are memory bound and waiting due to high memory latency.

D. We copy about 2KB into shared memory as we have a tile size of 16 in our blur kernel and we are storing integers which are four bytes long. We could have possibly changed the tile size to 32 and computed more parts of the array as that would also be able to fit into 8KB of shared memory.

4 Plots

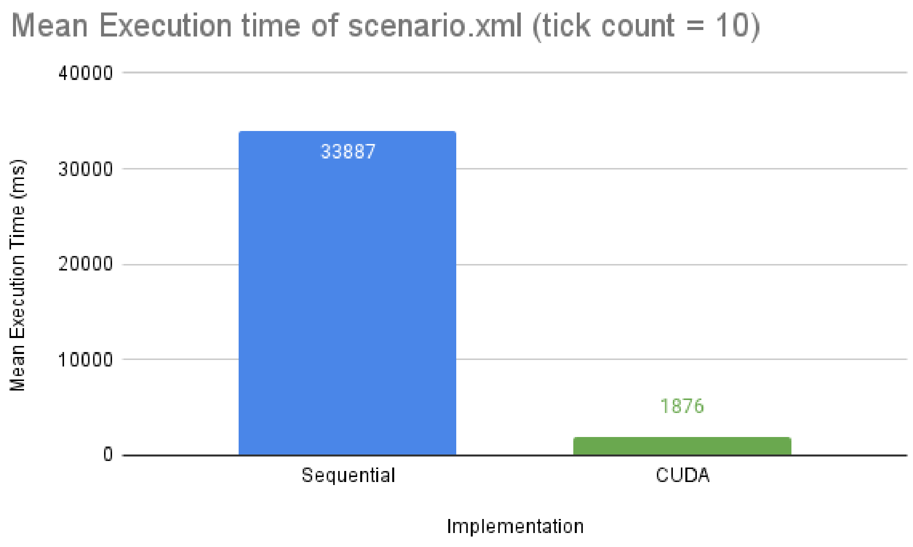

Figure 1: Mean Execution Times of scenario.xml

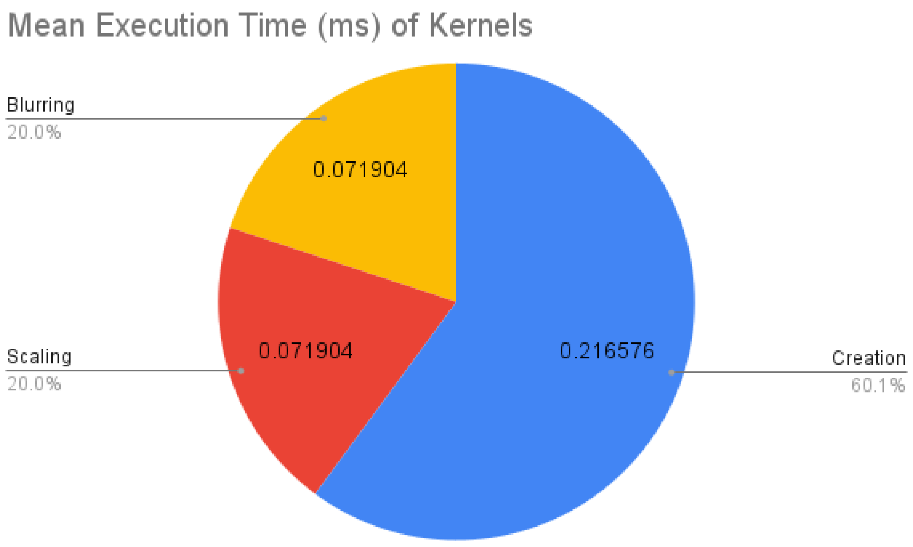

Figure 2: Mean Execution Time (ms) of heatmap computation steps

5 Contribution

Arveen Emdad, Hamit Efe Çınar, and Iason Kaxiras have contributed equally to the solution of this assignment.

References

[1] *Snowy User Guide*. 2023. URL: `https://www.uppmax.uu.se/support/user-guides/snowy-user-guide/`. (accessed: 2024-01-22).