Name: Rohit Pujar Nagraj
USC ID: 4664 4939 18

Programming Homework 6 - Report

Execution Times

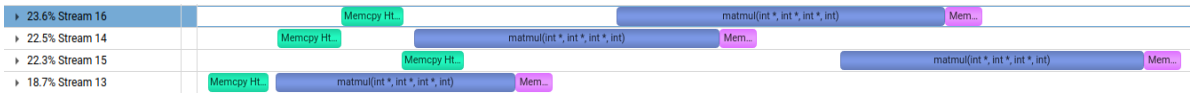
	nStreams = 1	nStreams = 4	nStreams = 16
Approach 1	15.671264ms	12.742656ms	11.988800ms
Approach 2	15.721440ms	14.694400ms	14.693376ms

C[451][451]: 208,282,624

Platform:

GPU: RTX 3090
CUDA version: 12.6
Number of copy engines: 2

P1 Execution Timeline



P2 Execution Timeline

Observations

Execution times: As we can observe from the above execution timelines captured by nsight system, we observe that the kernels execute in parallel to the HtoD data transfer in approach 1. However, in approach 2, the kernels on each stream wait for the HtoD data transfer to complete across all streams before beginning execution, which causes longer execution times for approach 2.
The reason we observe this performance difference is due to a concept called Lazy Loading in CUDA wherein the kernel is not loaded into memory until called. Turning off lazy loading as described here makes approach 2 run just as fast as approach 1 as quoted here which states that on newer compute versions, the order of launch does not impact the performance (Source).

Name: Rohit Pujar Nagraj
USC ID: 4664 4939 18

nStreams: Since in approach 1, kernel execution happens in parallel to the data transfer, we observe performance improvement as we increase n streams. However, increasing n to be too high will result in a performance drop due to the overhead of creating streams. Approach 2 also benefits a little from adding more streams as the kernel executions themselves are in parallel.

Output Screenshot