Lab 4 Report

Quan Fan
862099688
qfan005@ucr.edu

Questions

1. What is the execution time of your vector add using no streams and 3 streams?

Answer:

The time from copying data to device(s) to copying data back to host:

1 stream: 0.001775 s

3 streams: 0.001453 s

2. Were you able to observe full overlapping of computation and memory transfer? (Please use profiler output to justify your answer). Why or why not?

Answer:

Yes. The HtoD copying kept busy. All computations were overlapped by memory transfer except for the final launched kernel.

The output of profiler:

Start	Duration	Stream	Name
394.91ms	180.87us	17	[CUDA memcpy HtoD]
395.10ms	185.95us	17	[CUDA memcpy HtoD]
395.30ms	192.39us	19	[CUDA memcpy HtoD]
395.31ms	34.784us	17	VecAdd(int, float const *, float const *, float*) [434]
395.36ms	190.05us	17	[CUDA memcpy DtoH]
395.49ms	188.80us	19	[CUDA memcpy HtoD]
395.69ms	30.080us	19	VecAdd(int, float const *, float const *, float*) [436]
395.69ms	186.79us	18	[CUDA memcpy HtoD]
395.73ms	184.74us	19	[CUDA memcpy DtoH]
395.89ms	179.87us	18	[CUDA memcpy HtoD]
396.08ms	27.712us	18	VecAdd(int, float const *, float const *, float*) [435]
396.11ms	179.39us	18	[CUDA memcpy DtoH]

3. How does your profiler timeline look now? Are you able to observe full overlapping of computation and memory transfer?

Answer:

No. Only the first launched kernel was fully overlapped by memory transfer.

The output of profiler:

Start	Duration	Stream	Name
364.69ms	180.35us	17	[CUDA memcpy HtoD]
364.87ms	185.70us	17	[CUDA memcpy HtoD]
365.07ms	189.28us	19	[CUDA memcpy HtoD]
365.07ms	591.30us	17	VecAdd(int, float const *, float const *, float*) [434]

365.27ms	187.97us	19	[CUDA memcpy HtoD]
365.47ms	188.32us	18	[CUDA memcpy HtoD]
365.60ms	616.55us	19	VecAdd(int, float const *, float const *, float*) [436]
365.66ms	185.80us	18	[CUDA memcpy HtoD]
365.67ms	185.54us	17	[CUDA memcpy DtoH]
366.15ms	633.77us	18	VecAdd(int, float const *, float const *, float*) [435]
366.22ms	181.35us	19	[CUDA memcpy DtoH]
366.79ms	180.19us	18	[CUDA memcpy DtoH]

4. Would a Vector Add implementation with 4 streams help improve performance?

Answer:

No. 4 streams took 0.001566 s to finish.