COMP429 Project 1

Batuhan Başkaya 64240

Assignment

Part1

In this part, only data parallelism is usefully achievable since only 1 frame is generated and the MarchCube() function is called once. Inside this function, 3 other functions are called and their outputs are used as inputs on the next function. This prevents task parallelism in MarchCube(). We can't do something similar to in class summation example since the mesh could not be splitted like a sum variable. Since there are 3 dimensions worked on, we can paralellize on these dimensions(data parallelism). I paralellized on the X axis.

The minimum resolution I get 2x performance is 160 resolution with 4 threads. This drops the 3 seconds serial to 1.5 seconds.

I was not able to achieve perfect scaling because of non-parallelizable code, the thread creation overhead and critical section waiting time. For some reason, my code for part1 is getting extremely slow after 16 threads.

Part2

Since there are multiple frames now, I paralellized on these frames where each thread computes 1 frame at a time(data parallelism). Each thread does the same kind of computation, but for a different frame with a different twist.

The minimum resolution I get 2x performance is 20 resolution with 4 thread. This is much better than the first approach.

I was not able to get perfect scaling again because of thread creating overhead and the non-parallizable code section.
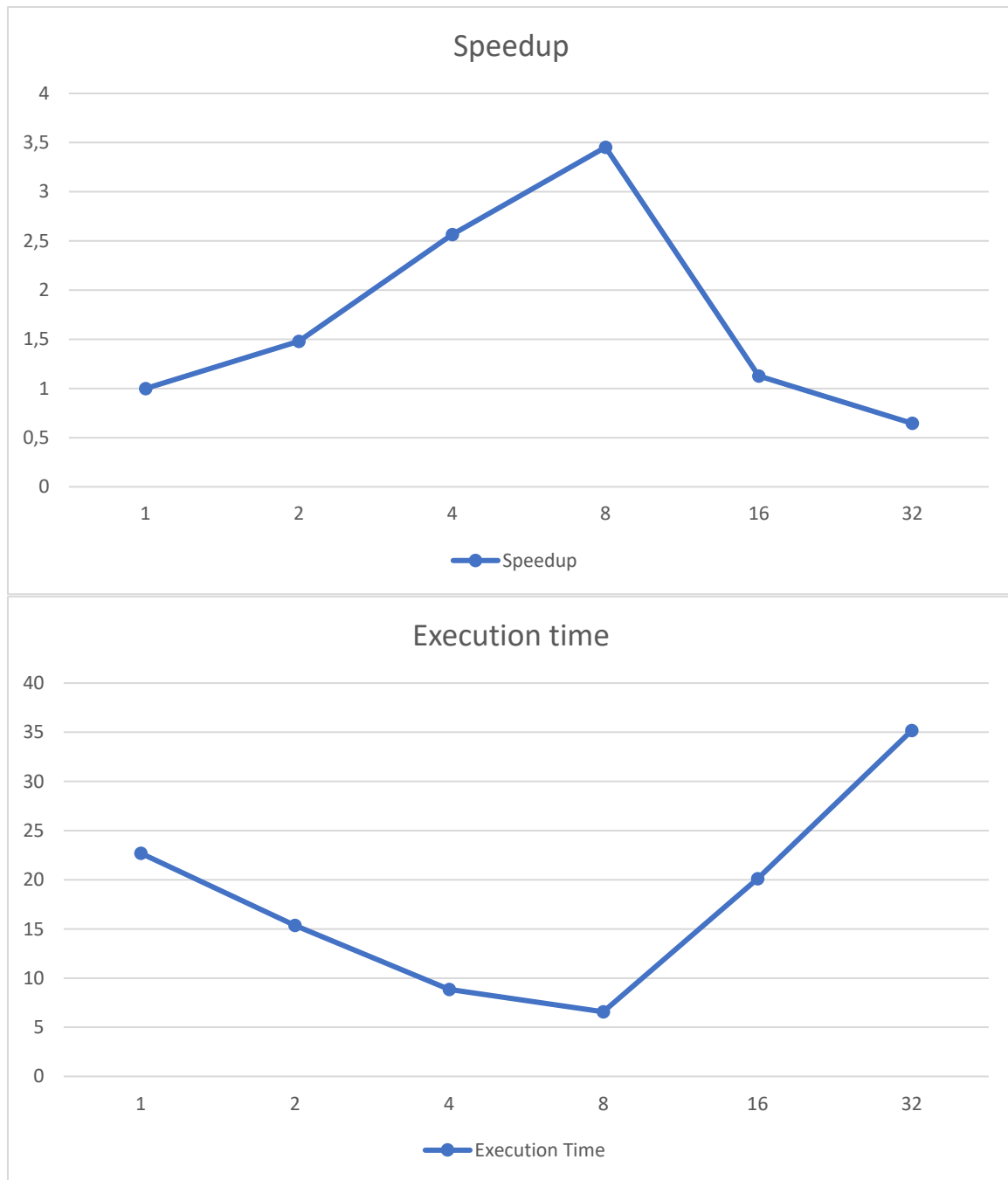
Part3

In this part, using only the 2nd approach will be more benefitial than to combine both of the approaches, since part1 was more slow in general and problematic. Part2 answers apply to the questions in part3.
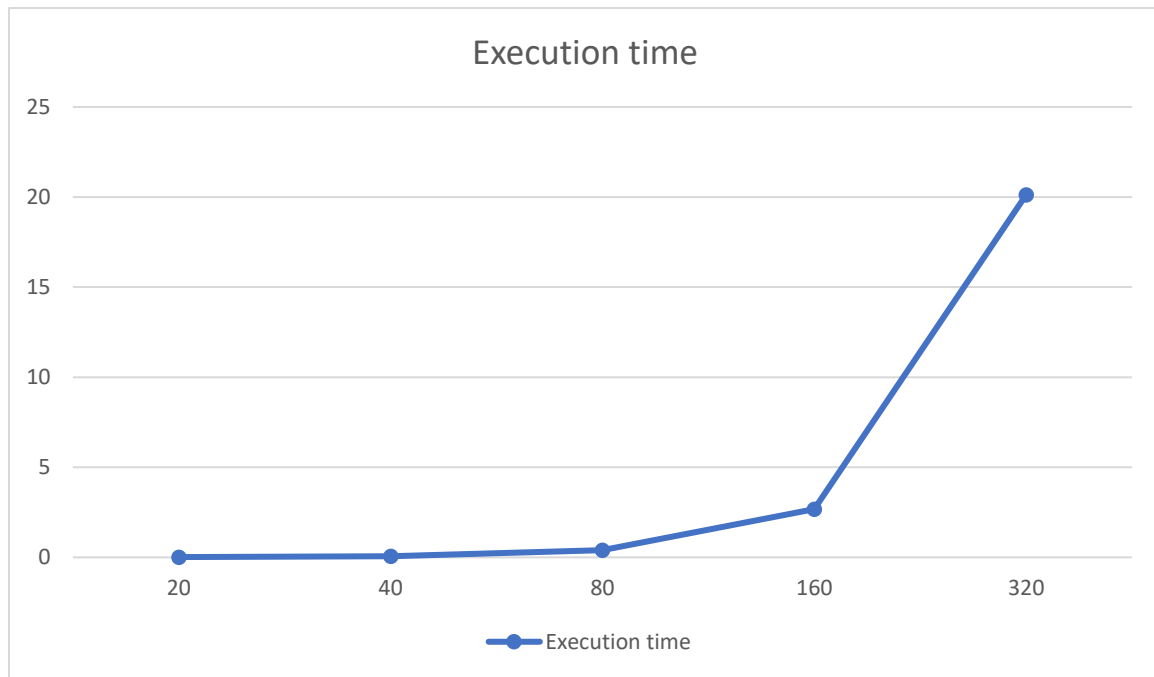
Experiments

Part1

I ran each case 4 times, and ignored the 1st run's result. I took the average of the last 3 runs.
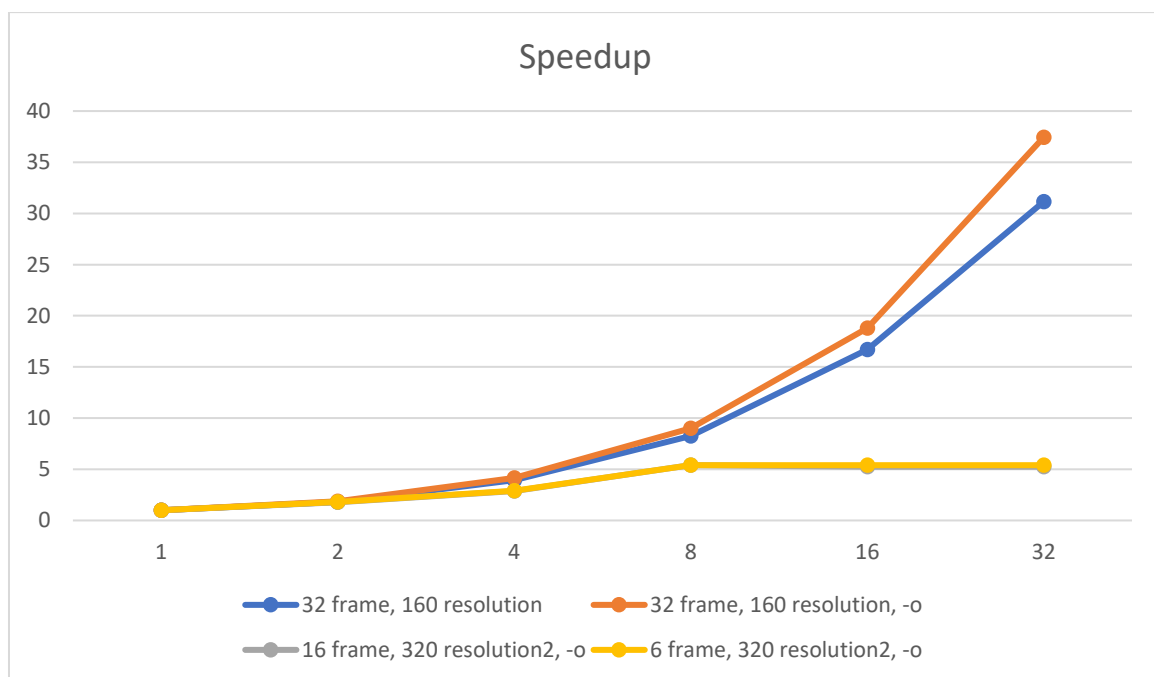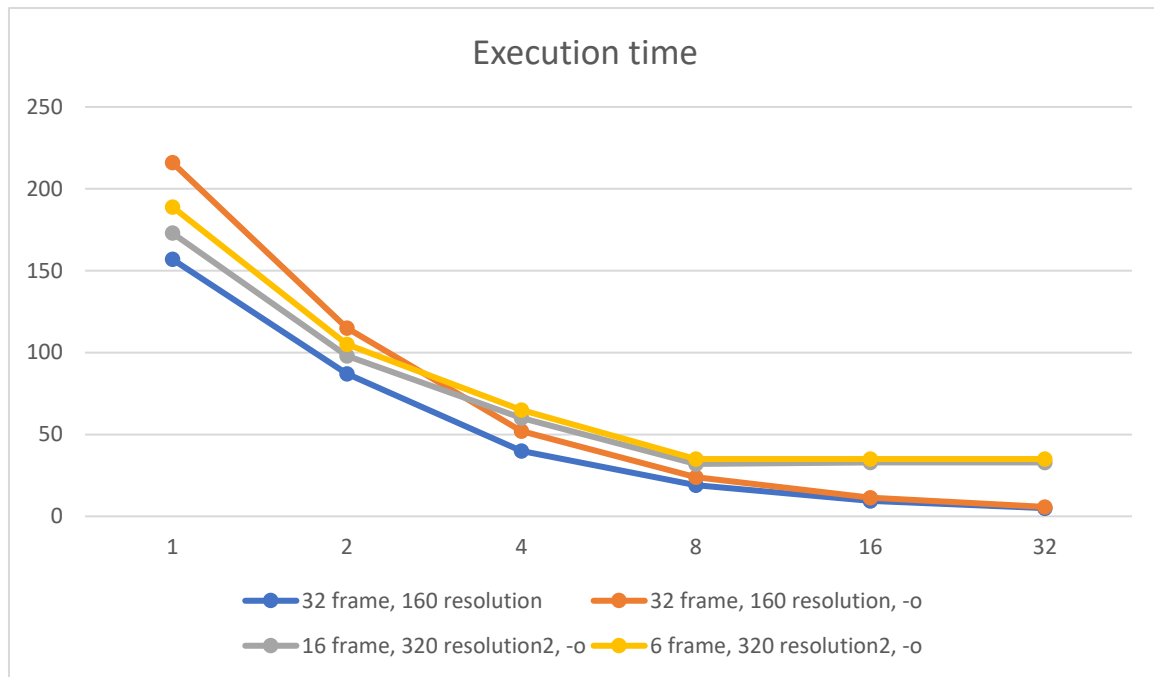
1st experiment

Speedup

Execution time

2nd experiment

Execution time

Part2/Part3(Same code)
1st experiment

Speedup

Execution time

2nd experiment

Execution time (10 frames)

Execution time (80 resolution)