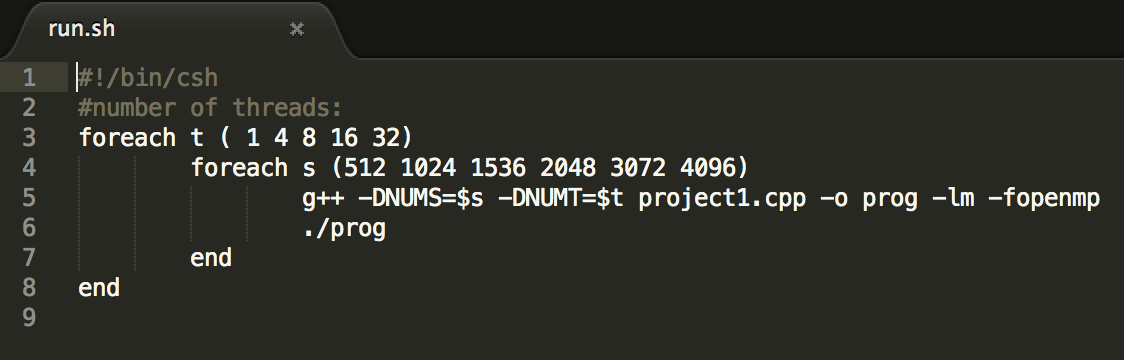
Parallel Programming

CS575

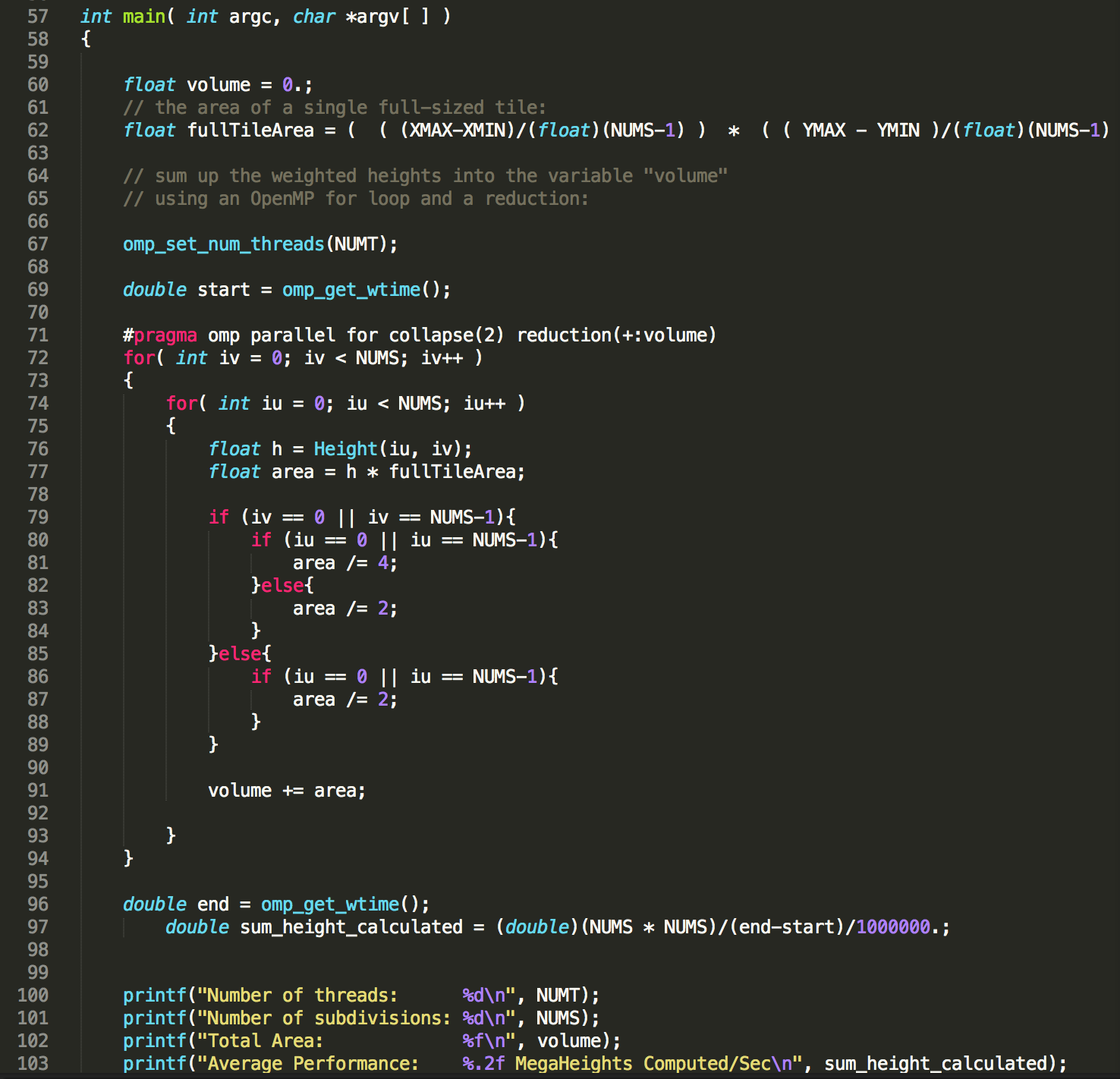
Chao Zhang

Project #1

1. Source listing



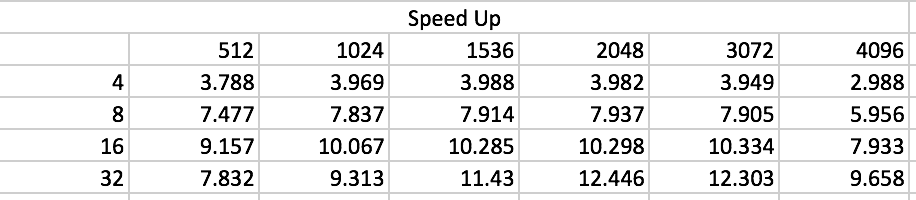
This is the run.sh file, it is used to run the program file which is the project1.cpp.With this .sh file, I can get all the numbers I need at one time. The NUMT are 1, 4, 8, 16 ,32 and the NUMS are 512, 1024, 1536, 2048, 3072 and 4096.



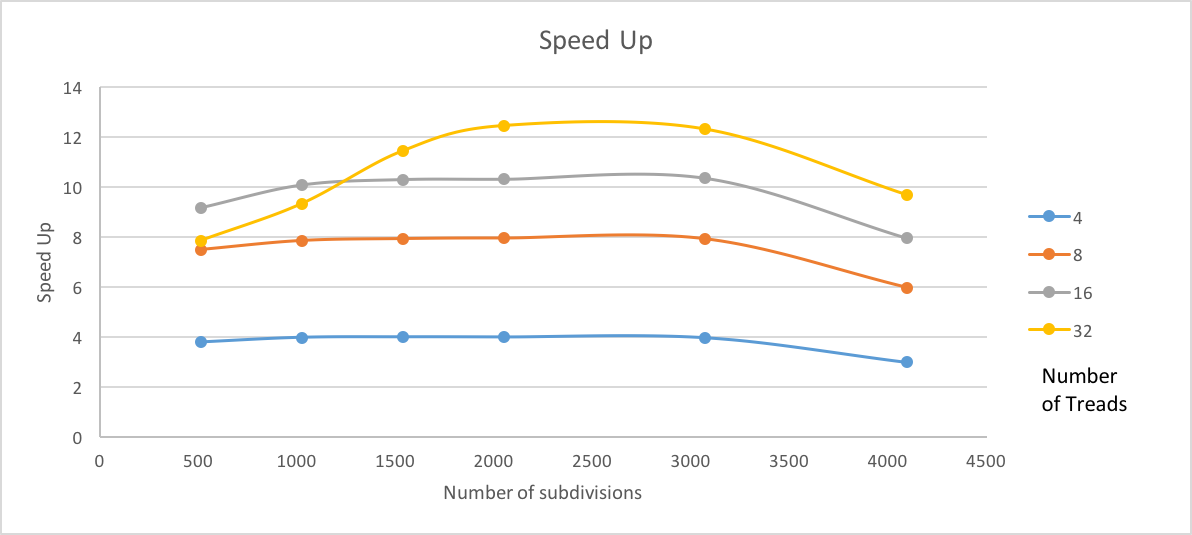
This is the most important part of this project, it uses the OpenMP for loop and the reduction to make sure the results is correct. Inside the for loop, it will calculate the volume use the area and the iu and iv.

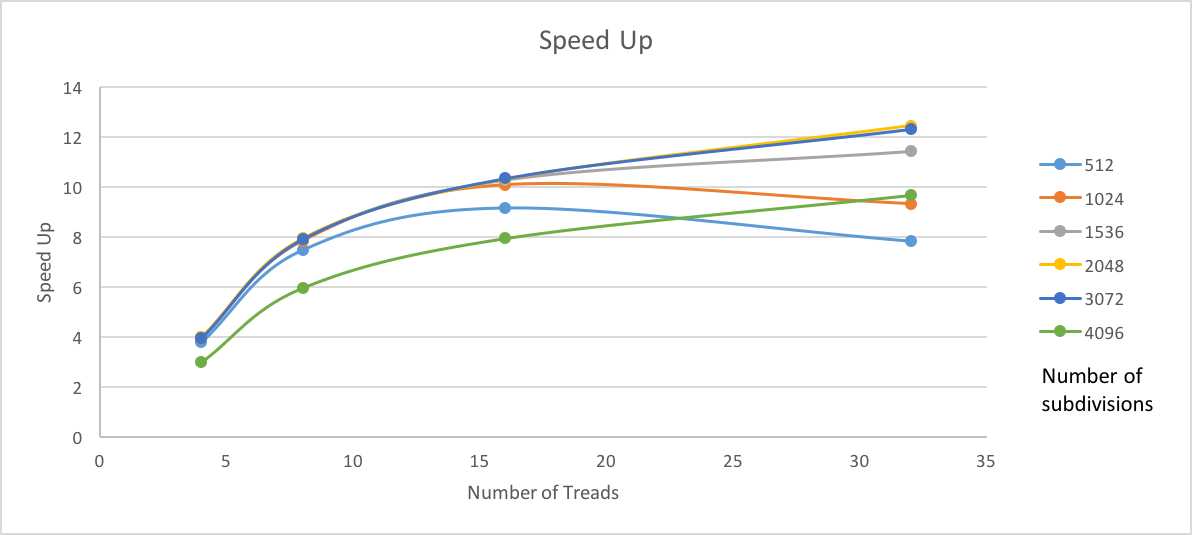
1. Result and analysis

I will use the graphs to show the final results. The speed up is the time for 1 core/ time for n core.



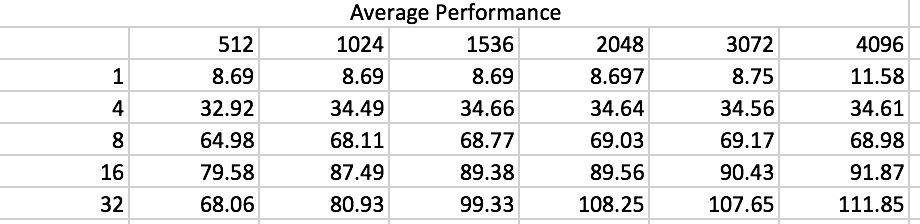
Speed Up Table



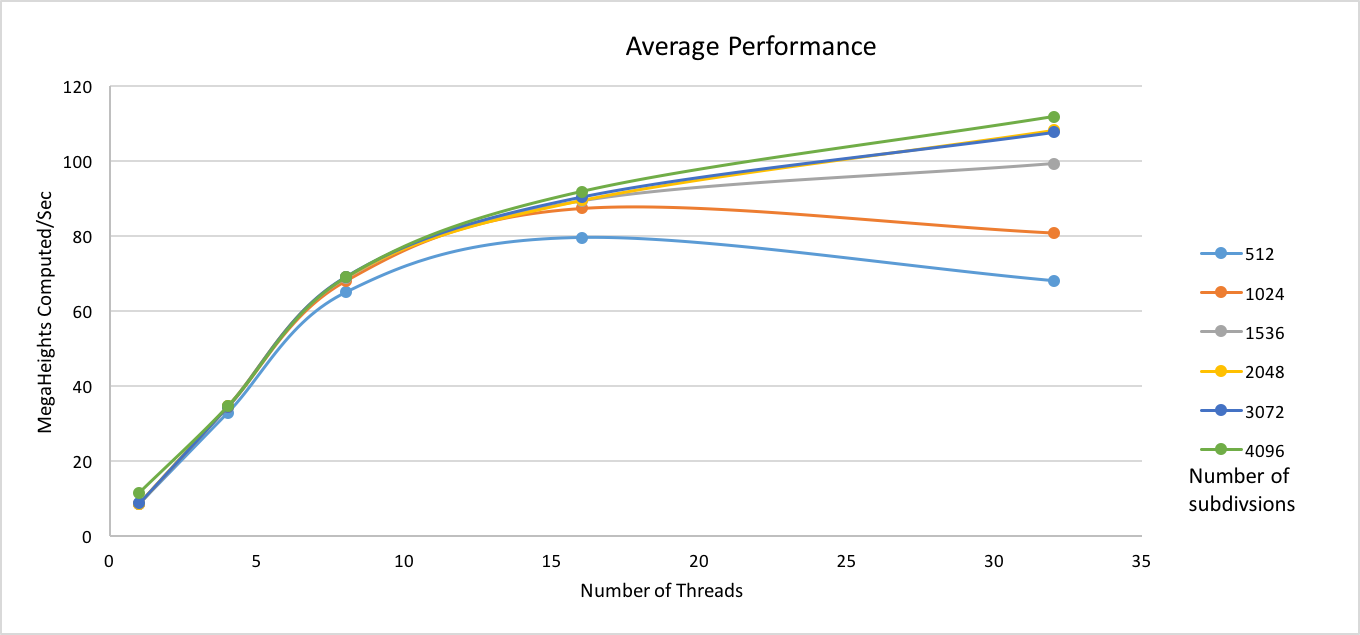
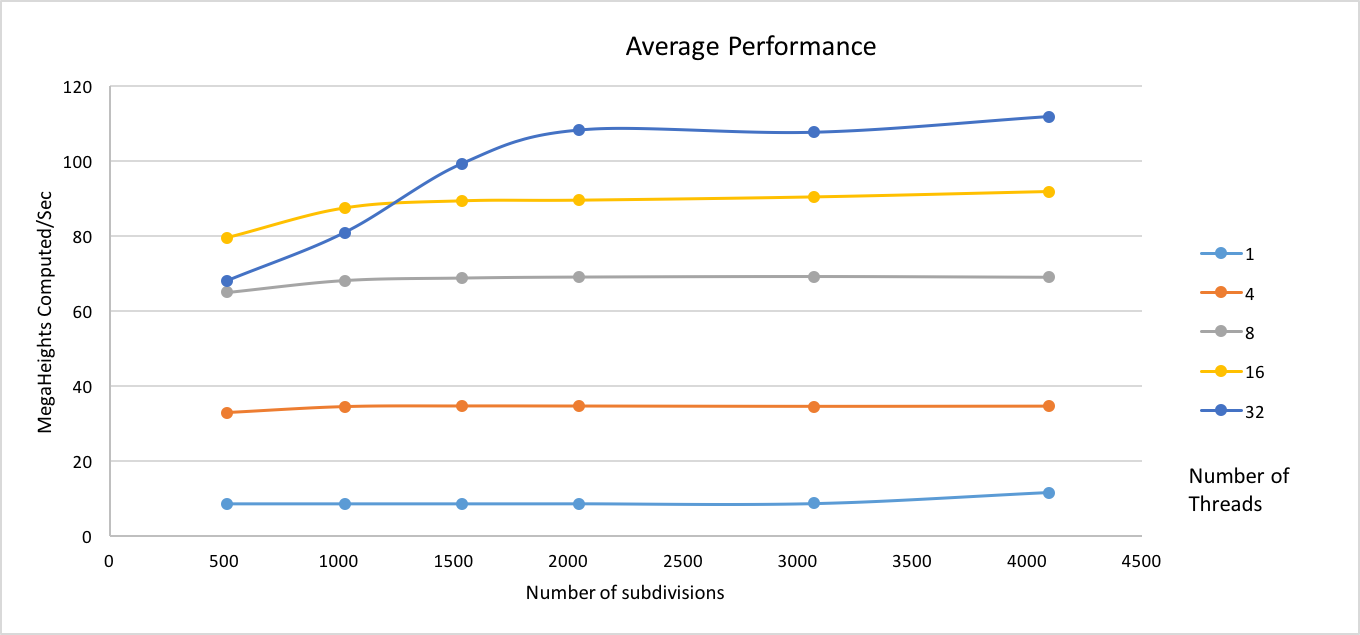


Speed Up Graphs

The average performance is the MegaHeights Computed/Sec of the project.

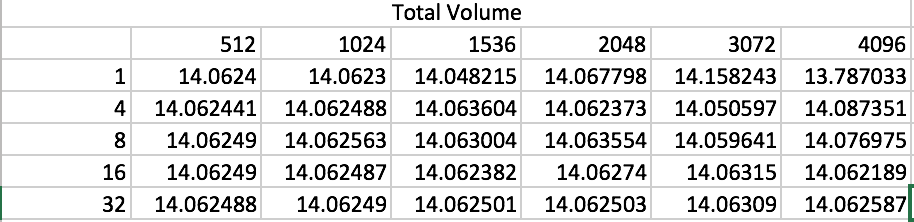


The Average Performance Table



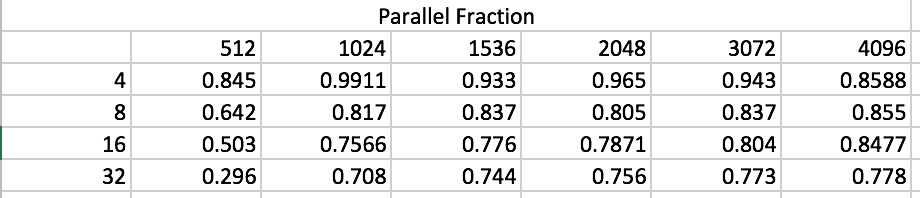
The Average Performance Graphs

The reason for the MegaHeights Computed/Sec of the 32 treads with 512 subdivisions is lower than the 16 treads is because the total of subdivisions is lower than the capability of the 32 treads. It finished quickly so the MegaHeights Computed/Sec is lower. The same reason for the speed up.

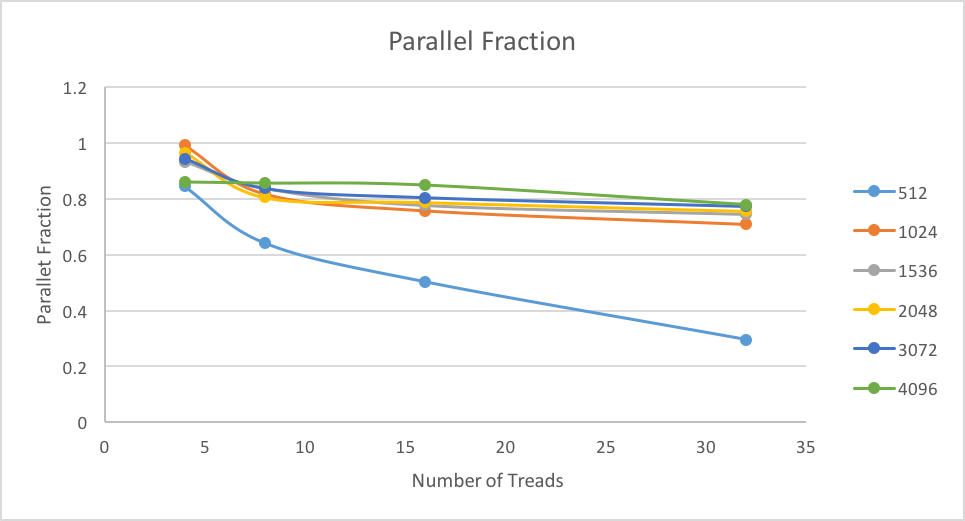


With all the data I have, the total area which is the volume are around the 14.065, so I think the actual volume is 14.065.

The Parallel Fraction is calculate with the formula F = NUMT / (NUMT – 1) \* (1 – 1 / speedup).



Parallel Fraction Table



Parallel Fraction Graphs

The parallel fraction is going down with the increase of the threads, this is because the fraction parallel means the percentage of runtime of the part that can be parallelized in the program. With the growth of threads, the runtime of parallelized part is reduced

The maxSpeedup = 1 / (1- F). The max F is 4 threads in 2048 subdivisions which is 0.965, so the max speedup is 28.57.