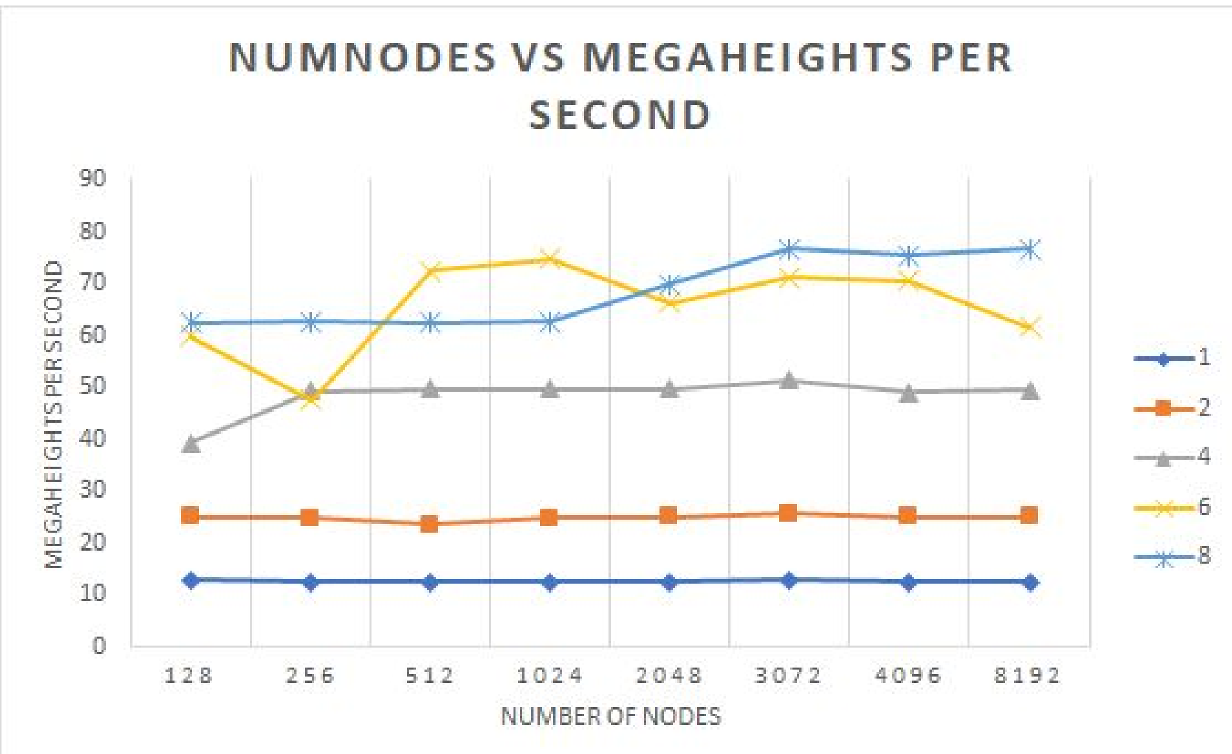
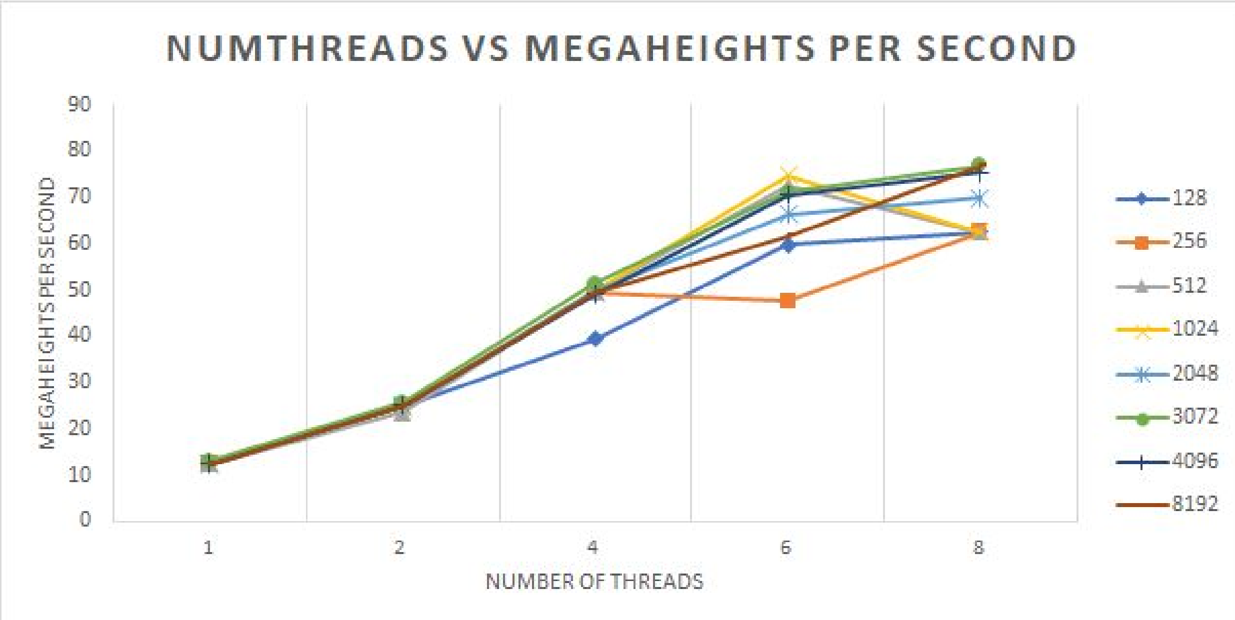
Coulby Nguyen
Project 2 Write Up
4/28/2019

1. What Machine Did I run this on
    a. I compiled and ran this assignment on PuTTY and used an SSH to the Oregon State University Servers.
2. What is the expected volume.
    a. The expected volume from the simulations produced an expected volume of ~28.462 Units ^ 3.
3. Show the performances you achieved in tables and graphs as a function of
    a. Number of Nodes
        i. Using number of nodes as the independent variable, x axis, in the set of {128, 256, 512, 1024, 2048, 3072, 4096, 8192}
        ii. Using the megaheights that are calculated per second as the dependent variable, y axis.
        iii. Each line represents a different thread in the set of {1, 2, 4, 6, 8}.



    b. Number of threads
        i. Using the number of threads as the independent variable, x axis, in the set of {1, 2, 4, 6, 8}

ii. Using the megaheights that are calculated per second as the dependent variable, y axis.
iii. Each line represents a different node size in the set {128, 256, 512, 1024, 2048, 3072, 4096, 8192}



4. What patterns are you seeing in the speeds?
    a. There are clear patterns that the speed increases as the number of threads used increases. There is an exception when the thread was of size 6, in which it performed lower than threads of size 4, and better than threads of size 8. There is also a pattern in regards to the size of nodes, in which smaller sizes performed worse than the larger node sizes. This can be shown clearly when the the threads size was 6, the bottom 2 lines, were the 128 node size, and the 256 node size.
5. Why do you think it is behaving this way?
    a. I believe the behavior of smaller node sizes performing worse than larger sizes is because of the Gustafson's observation that with larger datasets the parallel fraction increases. For the consistency of performance increasing by adding more threads is because that this is a problem that can be easily broken apart for parallelism and by adding more threads the data is partitioned into smaller sizes and each one is ideally completed in less time.
6. What is the Parallel Fraction for this application using the Inverse Amdahl Equation
    a. $S = (MegaHeights_{4\ thread})/(MegaHeights_{1\ thread})$
    b. $S = 49.3994 / 12.4953$
    c. $S = 3.95$

d. $F_{parallel} = (4.0/3.0) * (1.0 - (1.0/S))$
e. $F_{parallel} = (4.0/3.0) * (1.0 - (1.0/3.95))$
f. $F_{parallel} = (1.3333) * (1.0 - 0.253)$
g. $F_{parallel} = (1.3333) * (.747)$
h. $F_{parallel} = .995$

7. Given that Parallel Fraction, what is the maximum speed-up you could *ever* get
    a. $Max\ Speedup = 1 / (1 - F_{parallel})$
    b. $Max\ Speedup = 1 / (1 - .995)$
    c. $Max\ Speedup = 1 / (0.005)$
    d. $Max\ Speedup = 200$
8. Raw Data (Table)

Number of threads	Number of Nodes	Calculated Volume	MegaHeights Per Second
1	128	29.1449	12.928
1	256	28.916	12.496
1	512	28.8015	12.4975
1	1024	28.7436	12.4194
1	2048	28.756	12.4997
1	3072	28.6687	12.997
1	4096	28.8379	12.4543
1	8192	16.872	12.4953
2	128	29.1449	25.0363
2	256	28.9161	24.931
2	512	28.8018	23.5331
2	1024	28.7451	24.7914
2	2048	28.7419	24.991
2	3072	28.6976	25.7416
2	4096	28.7219	24.992
2	8192	25.4096	24.9856
4	128	29.1448	39.2925
4	256	28.9161	49.2044
4	512	28.8018	49.7184
4	1024	28.7447	49.6946
4	2048	28.7119	49.7398

4	3072	28.7037	51.4446
4	4096	28.6881	49.0741
4	8192	30.0125	49.3994
6	128	29.1449	59.8386
6	256	28.9161	47.5752
6	512	28.8018	72.4361
6	1024	28.7447	74.717
6	2048	28.7168	66.364
6	3072	28.7081	71.0726
6	4096	28.7375	70.4596
6	8192	28.6674	61.6229
8	128	29.1449	62.44
8	256	28.9161	62.6307
8	512	28.8018	62.5571
8	1024	28.7447	62.6272
8	2048	28.7157	69.9596
8	3072	28.6982	76.7794
8	4096	28.7355	75.4104
8	8192	28.6546	76.7473