1. Tell what machine you ran this on

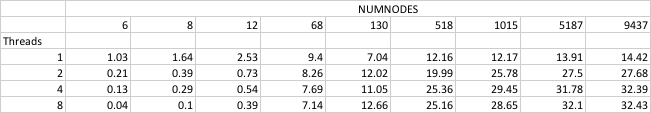
MacOS X Yosemite Version 10.10.5

* 1. GHz Intel Core i7, 4 GB

1. What do you think the actual volume is?

25.

1. Show the performances you achieved in tables and graphs as a function of NUMNODES and NUMT



1. What patterns are you seeing in the speeds?

From Chart #1, we conclude that as the number of threads increase, performance also increases but since we do not require more computation, the performance of 4th thread is approximately similar to that of 8th thread.

From Chart #2, we conclude that when the number of nodes is 68, the performance is almost constant even after increasing the threads. But as we give more data to compute, we get better performance(parallelism) which can be inferred from the performance obtained by computing 5187 nodes. [Gustafson’s Observation]

1. Why do you think it is behaving this way?

We know that performance is determined by square of number of nodes divided by execution time. So as we increase the number of threads the execution time decreases and thus the performance increases.

1. What is the Parallel Fraction for this application using the Inverse Amdahl equation?

Speedup = P8/P1

= 32.43/14.42 = 2.25

Float Fp = (8.0/7.0) \* (1.0-(1.0/S))

= (1.14) \* (1.0- (1.0/2.25))

= (1.14) \* (1.0- 0.44)

= (1.14) \* (0.56)

= 0.64

1. Given that Parallel Fraction, what is the maximum speed-up you could ever get?

Max Speedup = 1/(1-Fp)

= 1/ (1-0.64)

= 1/0.36

= 2.8