PROJECT 2: Numeric Integration with OpenMP

1. **Tell what machine you ran this on.**

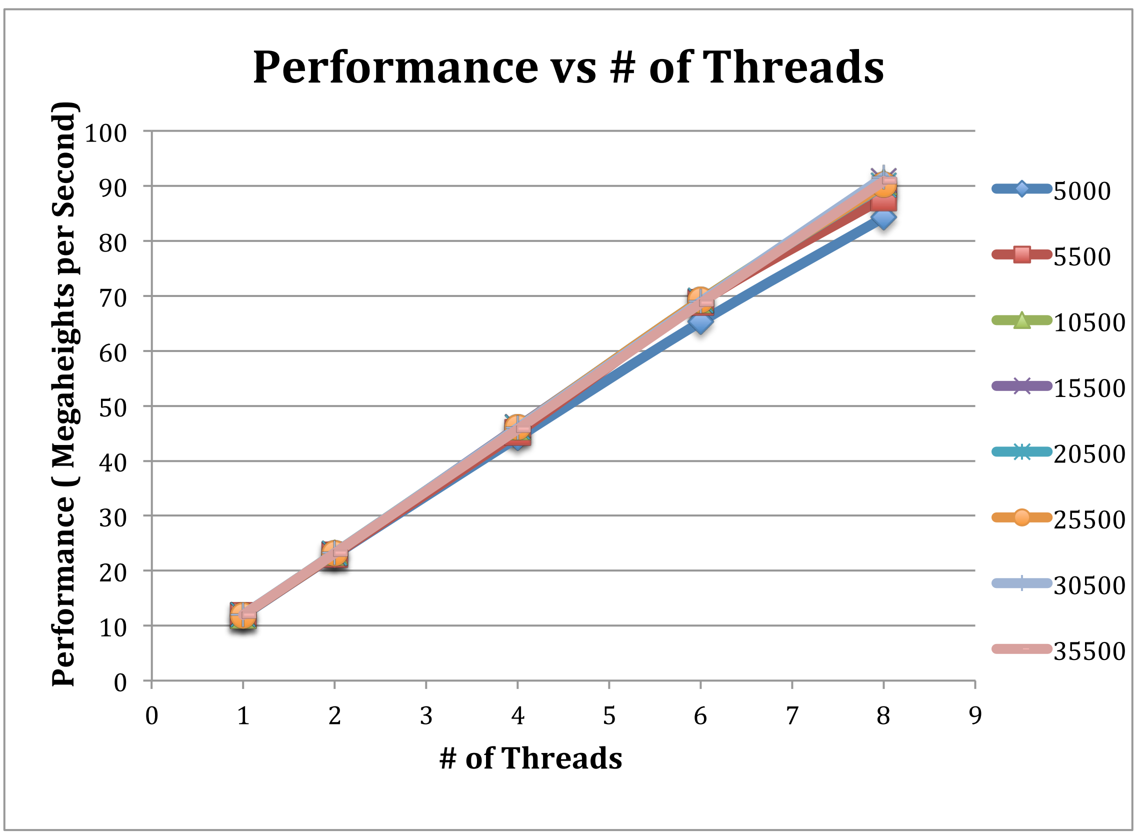
    - I ran this on the OSU flip machine.

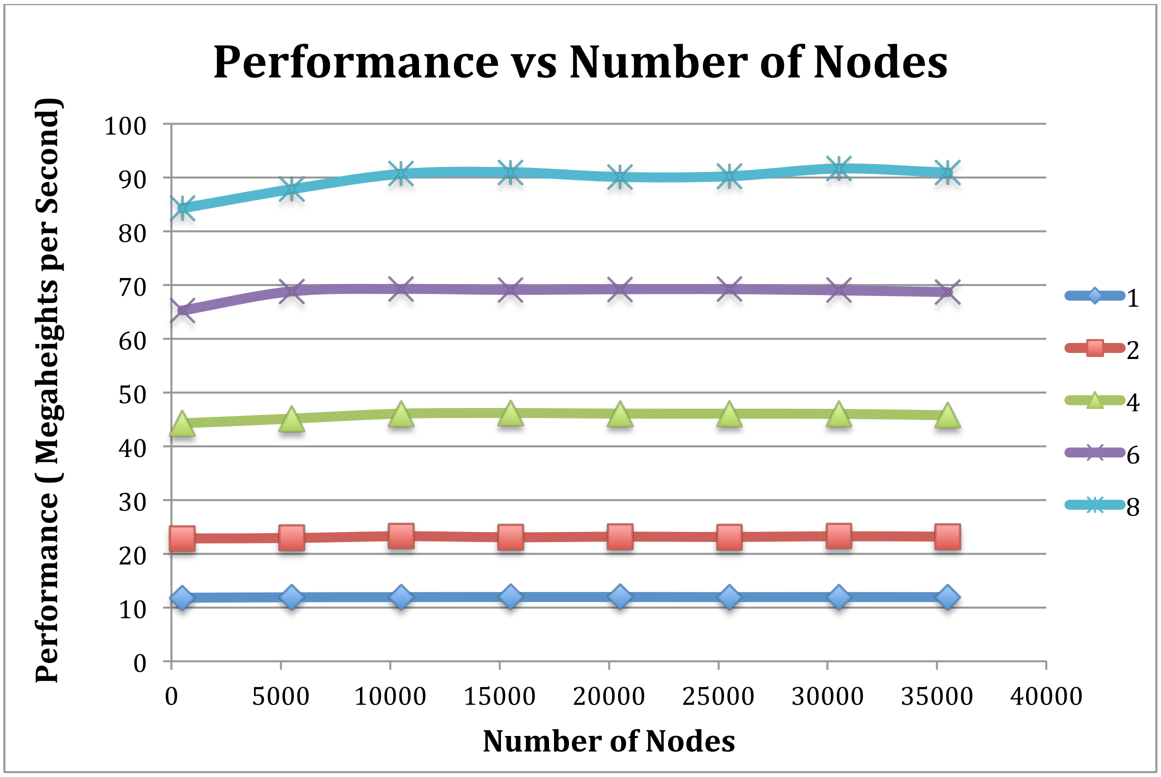
2. **What do you think the actual volume is?**

    - The actual volume that I got = **28.69**

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

NUMT	NUMNODES							
	500	**5500**	**10500**	**15500**	**20500**	**25500**	**30500**	**35500**
1	11.82	11.91	11.94	11.96	11.96	11.94	11.95	11.94
2	22.86	22.94	23.27	23.08	23.19	23.14	23.27	23.19
4	44.26	45.16	46.08	46.2	46.09	46.07	46.04	45.78
6	65.29	68.84	69.29	69.12	69.24	69.25	69.03	68.71
8	84.31	87.85	90.7	91.02	90.13	90.26	91.7	90.93





4. **What patterns are you seeing in the speeds?**

    - As the Number of threads increase, the performance improves.
    - As the Number of nodes increase, the performance improves but caps at a certain point and doesn't really significantly change after that.

5. **Why do you think it is behaving this way?**

    - As the # of cores increases, the execution time of parallel portion decreases but that of sequential portion remains the same. The Sequential potion doesn't go away and it doesn't get any smaller, just gets more and more dominant. This is why after a certain point, the performance doesn't improve or significantly change after a while. There is always some fraction of the total operation that is inherently sequential and cannot be parallelized no matter what we do.

6. **What is the Parallel Fraction for this application, using the Inverse Amdahl equation?**

- First, lets calculate

  Speedup = P_8/P_1

  = 90.93/11.94 = **7.615**

  Now calculating parallel fraction,

  $F_p = \left(\frac{n}{n-1}\right)\left(1-\left(\frac{1}{Speedup}\right)\right)$ = (8/(8-1)) * (1- (1/7.61)) = **0.992 = approx. 0.99**

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

- Maximum speedup = $1/(1-F_p)$ = 1/(1-0.99) = **100**