Aashutosh Taikar
Student ID:933020005
CS575: Project1

NUMERIC INTEGRATION WITH OPENMP

1. Tell what machine you ran this on

   Ubuntu running on Intel Core i5 8GB RAM

2. What do you think the actual volume is?

   Volume computed = 25.32

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

	NUMNODES							
no. of threads	4	10	50	70	100	500	700	1000
1	1.67	2.24	2.55	2.55	2.54	2.52	2.57	2.58
2	0.57	1.91	4.69	4.7	4.14	4.93	4.82	4.87
4	0.18	1.15	7.06	0.4	8.64	8.38	6.47	7.67

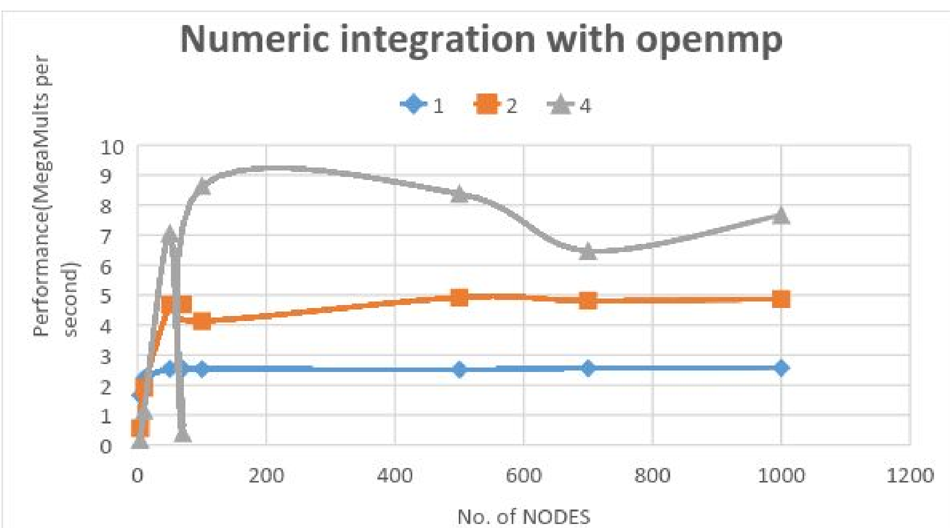

figure 1

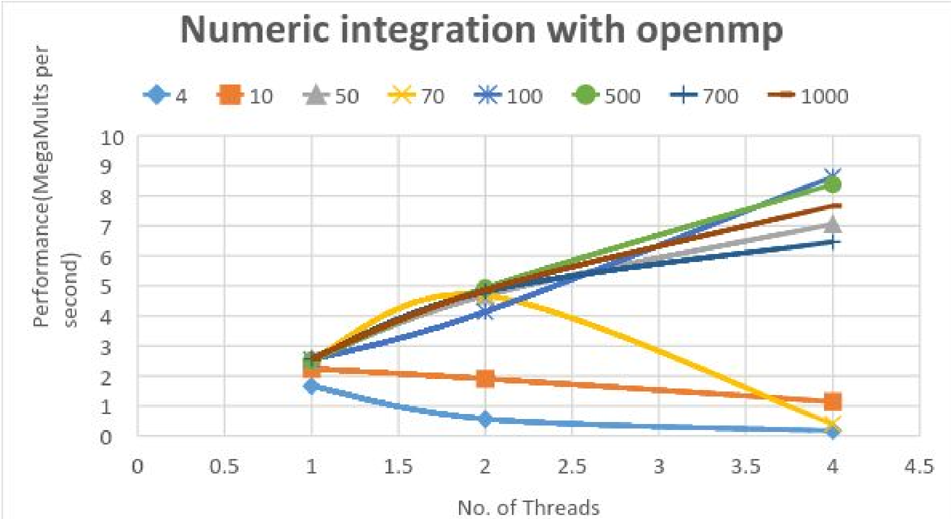

figure 2

4. What patterns are you seeing in the speeds?

As the number of threads increase the performance of integration increases. It is seen from the grey line in figure.1. as using 4 number of threads outperforms as compared to using 1 thread or 2 threads.

From the figure 2 we get to know that the performance is low when using 1 thread and 1000 nodes, but also it is seen that to compute 4 nodes on 4 threads will give a low performance. Even if we increase the threads the performance will fall if the Thus, the number of nodes as well as the threads should be increased to achieve more performance.

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

   Amdahl's law explains that the serial fraction and the parallel fraction contributes to the overall speedup. This can be seen by figure 1 as we increase the number of threads to compute more nodes of the beizer surface the performance is increases.
   The performance is characterized by the number of data over the execution time. As we increase the threads, the execution time decreases and the performance increases. This is explained by Gustafson where this parallel

fraction is related to the amount of data to be computed given by NF. In our case when we compute more number of nodes of the beizer surface more parallel computations can be done using more number of threads or cores to improve the performance.

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

   Speedup (S):
   Consider the 4 thread to 2 thread performance while computing 1000 nodes
   =P4/P1
   =7.67/2.58
   =2.97

   Using Inverse Amdahl's equation:
   Fp = (n/n-1) x (1-1/S)
   = (4/3) x (1-1/2.97)
   = 0.88

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

MaxSpeedup = 1/(1-Fp)

= 1/(1-0.88)

= 8.33