CS-575 Parallel Programming

Spring 2022

Name: Akhil Sai Chintala

Email: chintala@oregonstate.edu

Project Name: #Project-2

1) **Tell what machine you ran this on?**

I have executed the project-2 code on the rabbit server "rabbit.engr.oregonstate.edu".

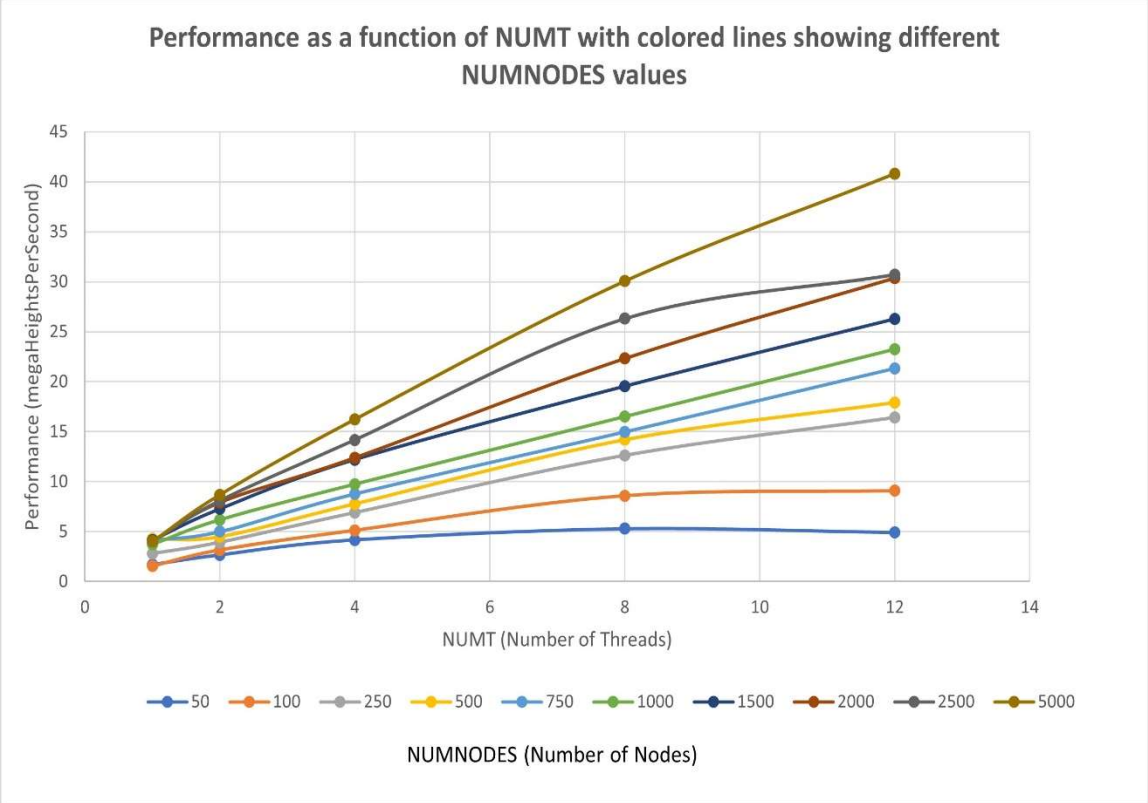
2) **What do you think the actual volume is?**

I think the actual volume is 7.75 cubic units.
This is because, the most common value for the volume among all the threads executed in the program is 7.75.

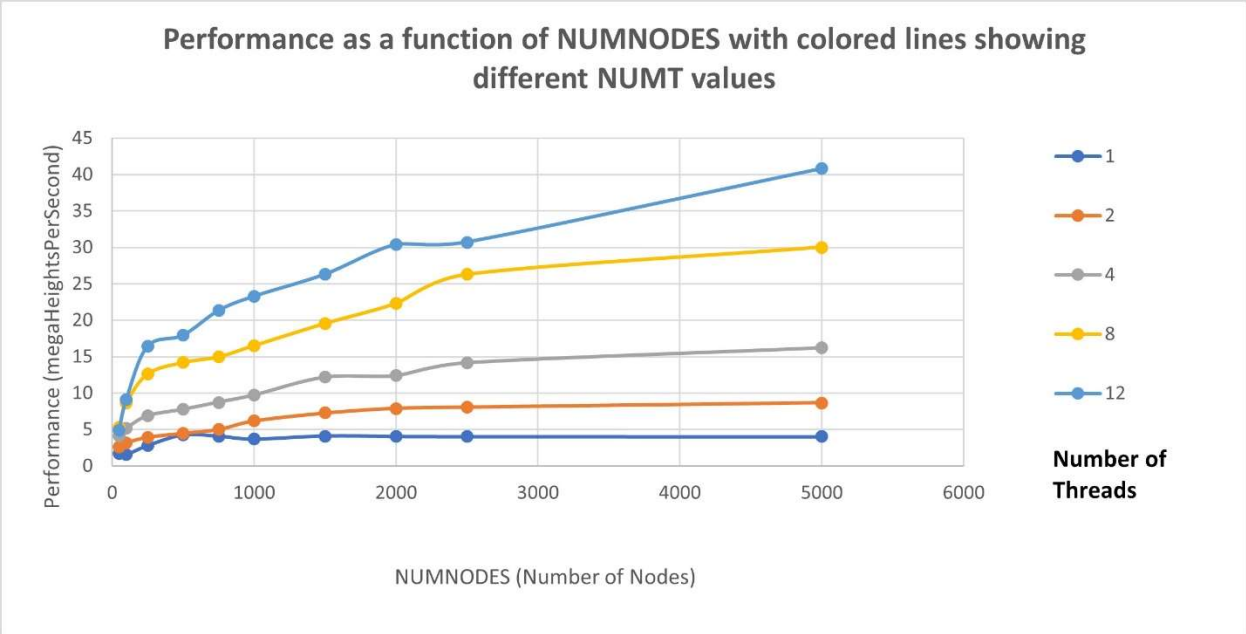
3) **Show the performances you achieved in tables and two graphs showing:**

Threads/NUMNODES	50	100	250	500	750	1000	1500	2000	2500	5000
1	1.71	1.56	2.8	4.28	4.12	3.71	4.14	4.07	4.04	4.02
2	2.68	3.17	3.93	4.49	5	6.18	7.27	7.93	8.09	8.7
4	4.18	5.14	6.9	7.78	8.77	9.75	12.21	12.4	14.18	16.24
8	5.29	8.6	12.62	14.2	14.97	16.52	19.56	22.32	26.31	30.06
12	4.92	9.1	16.4	17.92	21.31	23.27	26.3	30.37	30.72	40.8

a)



b)



4) What patterns are you seeing in the speeds?

Once the threads achieve the maximum value, I have observed that the performance of the threads are stagnated, and performance remains constant as the NUMNNODES are increased until the maximum value is reached.

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

I think this is because of the overloading of the tasks where the number of cores are not sufficient for the execution.

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

To calculate the parallel fraction (Fp), speedup needs to be calculated first. The Speedup formula is derived as follows,

$$\text{Speedup}(S) = \frac{\text{Performance of 12 threads (Maximum)}}{\text{Performance of 1 thread}}$$

(I am considering the performance for the maximum NUMNODES which is 5000).

= 40.8/4.02

= 10.149
Speedup = 10.149
Parallel Fraction Fp is calculated using speedup as below,
Fp = (n/n-1) * (1-1/s)
n-> maximum number of threads
s-> Speedup
Fp = (12/12-1) * (1-1/10.149)
= 0.983
Parallel Fraction (Fp) = 0.983.

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

The speedup of the respective threads and numnodes is given in the below table.

NUMNODES-→	50	100	250	500	750	1000	1500	2000	2500	5000
2 threads	1.567251462	2.032	1.4036	1.0491	1.2136	1.6658	1.756	1.9484	2.0025	2.1642
4 threads	2.444444444	3.295	2.4643	1.8178	2.1286	2.628	2.9493	3.0467	3.5099	4.0398
8 threads	3.093567251	5.513	4.5071	3.3178	3.6335	4.4528	4.7246	5.484	6.5124	7.4776
12 threads	2.877192982	5.833	5.8571	4.1869	5.1723	6.2722	6.3527	7.4619	7.604	10.149

After computing all the speedups, the maximum speedup is achieved by the 12 threads with 5000 NUMNODES with speedup as 10.149.

Maximum Speedup = 10.149