Name: Venkata Vasantha Phani Sai Paramkusham

Subject: CS_575

Project: Project 2

ONID: paramkuv

1. Tell what machine you ran this on?

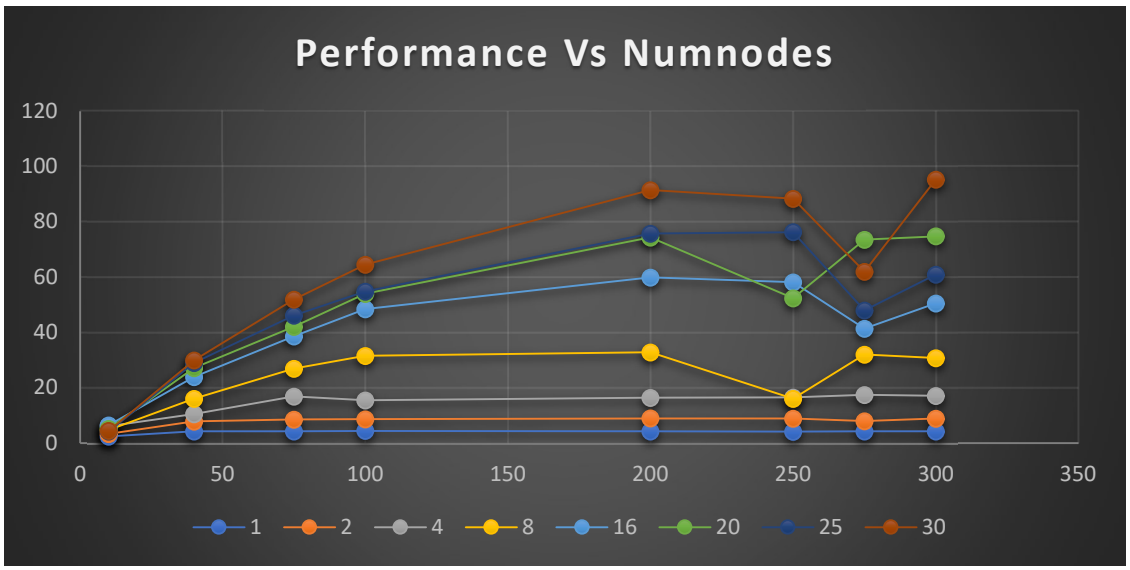
Ans. Rabbit

2. What do you think the actual volume is?

Ans. 7.5

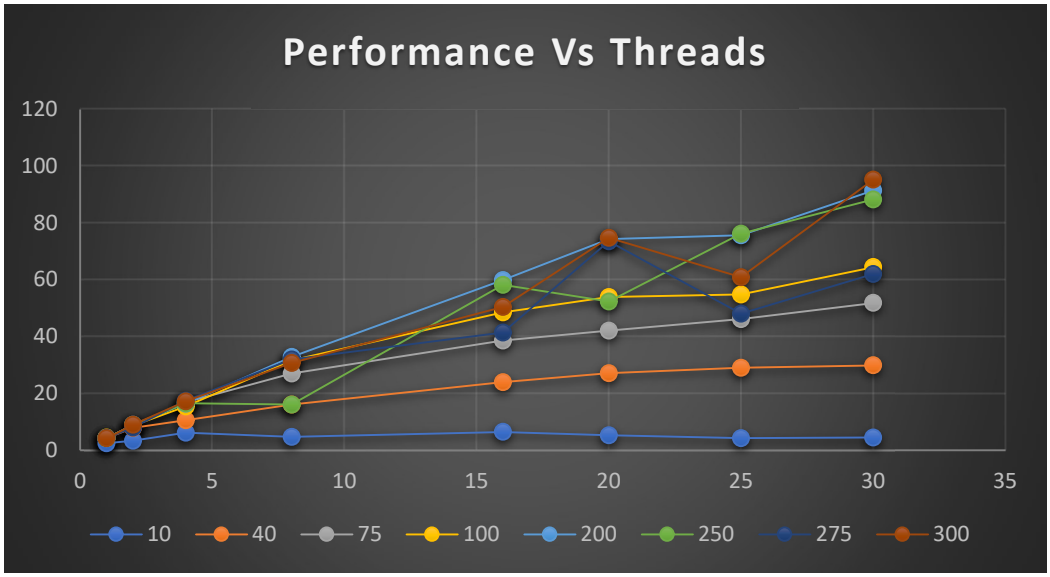
3. Show the performances you achieved in tables and two graphs showing:
    1. Performance as a function of NUMNODES with colored lines showing different NUMT values

    Ans.



2. Performance as a function of NUMT with colored lines showing different NUMNODES values

   Ans.



4. What patterns do you see in the speeds?

Ans) We can find in our analysis that even if we increase the number of nodes and threads the performance will be increased whereas with the increase in numnodes the time taken to execute the code also increases when we take threads as a constant. But in my graph speed in some regions got decreased due to an increase in the load of the system.

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

Ans) **Performance Vs. Numnodes Graph:**

The graph represents the Performance as a function Numnodes with respective Number of threads. In the Q2.1 graph, with the increase in the number of threads, the

performance of the graph is increasing. After a particular point, the graph becomes saturated. For example, after 100 nodes, it reaches saturation for eight threads.

But in my graph, after the Numnodes reach more than 250, the performances slightly dip. I feel this due to the load. In the system during I use the Rabbit system.

Even if we increase the threads sometimes “F-sequential” the performance will show some drop.

Performance Vs. Threads

The graph represents the Performance as a function NumT with respective NumNodes. In the graph, Q 2.2 graph was increasing with the increase in threads. But in my graph, when NumNodes are more than 250, my graph started showing the wrong results due to a change in load on the rabbit system.

In some scenarios, even though we increase the number of threads, the performance will not be improved.

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

Ans)

Speed up= performance of max thread/ performance min thread

= 95.341126/4.171304 = 22.856431945

PF = 30/29*(1-(4.171304/95.341126)) = 0.98922273023

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

Ans) Max Speed Up= 1/(1-Fp)= 1/(1-0.98922273023)= 92.7878786874

Miscellaneous:

Volume	Thread	NumNodes	Performance
6.653766	1	10	2.378851
7.529769	1	40	4.226498
7.63883	1	75	4.206186
7.668149	1	100	4.386629
7.713607	1	200	4.163233
7.722766	1	250	4.126963
7.725908	1	275	4.248473
7.728526	1	300	4.171304
6.653766	2	10	3.269367
7.529769	2	40	7.819143
7.63883	2	75	8.421528
7.668149	2	100	8.620579
7.713607	2	200	8.894379
7.722766	2	250	8.815641
7.725908	2	275	7.92659
7.728526	2	300	8.875198
6.653766	4	10	6.139
7.529769	4	40	10.467998
7.63883	4	75	16.791886
7.668149	4	100	15.379573
7.713607	4	200	16.376872
7.722766	4	250	16.483677
7.725908	4	275	17.410309
7.728526	4	300	17.099833
6.653766	8	10	4.686782
7.529769	8	40	16.007481
7.63883	8	75	26.906567
7.668149	8	100	31.557948
7.713607	8	200	32.833393
7.722766	8	250	15.971747
7.725908	8	275	31.920498
7.728526	8	300	30.653207
6.653766	16	10	6.39627
7.529769	16	40	23.913073
7.63883	16	75	38.625042
7.668149	16	100	48.444434
7.713607	16	200	59.877989
7.722766	16	250	58.154446
7.725908	16	275	41.375779
7.728526	16	300	50.457173
6.653766	20	10	5.159988

7.529769	20	40	27.066847
7.63883	20	75	42.067197
7.668149	20	100	53.961892
7.713607	20	200	74.346817
7.722766	20	250	52.364183
7.725908	20	275	73.55985
7.728526	20	300	74.79096
6.653766	25	10	4.150609
7.529769	25	40	29.013433
7.63883	25	75	46.104271
7.668149	25	100	54.805955
7.713607	25	200	75.714463
7.722766	25	250	76.269823
7.725908	25	275	48.007481
7.728526	25	300	60.944075
6.653766	30	10	4.371201
7.529769	30	40	29.859598
7.63883	30	75	51.847763
7.668149	30	100	64.504108
7.713607	30	200	91.462848
7.722766	30	250	88.3281
7.725908	30	275	61.985056
7.728526	30	300	95.341126