Nicholas Skinner

CS 475

Spring 2019

**Project 2 - Numeric Integration with OpenMP**

**Platform**

Tests for this project were conducted on the Oregon State University Flip Server.

**Volume**

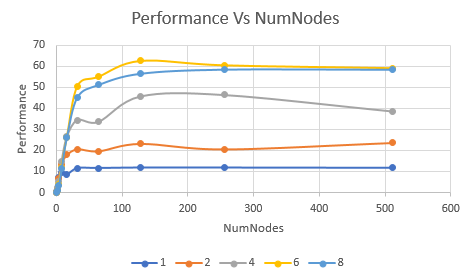
I believe the volume to be roughly **28.6875 units,** at least, that's what I ended up receiving from my program

**Table**

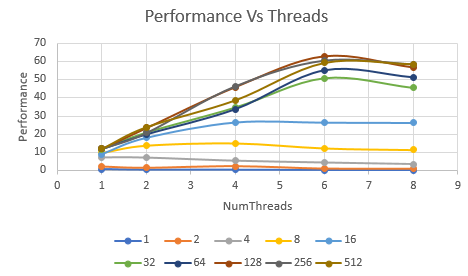
|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | NumNodes |  |  |  |  |  |  |  |  |  |
| Num Thread |  | 1 | 2 | 4 | 8 | 16 | 32 | 64 | 128 | 256 | 512 |
|  | 1 | 0.675253 | 2.21461 | 7.21579 | 9.88464 | 8.85198 | 11.7669 | 11.8312 | 12.0351 | 12.0155 | 11.8972 |
|  | 2 | 0.378428 | 1.53329 | 7.06453 | 13.687 | 18.0166 | 20.8355 | 19.8818 | 23.4325 | 20.7987 | 23.8804 |
|  | 4 | 0.344068 | 2.45554 | 5.25681 | 14.8712 | 26.6556 | 34.602 | 33.7517 | 45.8591 | 46.5095 | 38.6824 |
|  | 6 | 0.236751 | 1.07742 | 4.2339 | 12.2394 | 26.5342 | 50.7534 | 55.1635 | 62.7241 | 60.6232 | 59.2683 |
|  | 8 | 0.186593 | 1.02194 | 3.32747 | 11.3972 | 26.3915 | 45.4625 | 51.3245 | 56.6508 | 58.6021 | 58.5806 |

**Graphs**

**Performance vs NUMNODES**

****

**Performance vs NUMT**

****

**Patterns in speed:**

Through the above graphs, we can see that the lower NumNodes values will give reduced performance when using additional threads, this may be due to the overhead when the system is setting up, specifically 8 threads sees some poor performance at the 1 and 10 NumNodes values, but as the cycles went on, the 8 threads were able to display their abilities when even more NumNodes were thrown onto the pile.

The performance vs threads graph displays performance for individual NumNodes values for each thread level, an observation here is that above 6 threads we appear to be getting reduced performance. I think this may be due to the reduction, but I am uncertain if this is the real cause.

**Inverse ahmadal parallel fraction**

The parallel fraction for this setup

Speedup S = (results 4 threads / results 1 thread)

Amdahl’s Law states:

(n/(1-n)) \* (1-(1/S))

Where n = number of threads

And S = speedup

Using 6 threads and the 512 numnodes values, versus 1 thread 512 numnodes our equation should look like:

S = 59.2683/11.8972

S = 4.9817

And our parallel fraction should look like:

Fp = (6/5) \* (1-(1/4.9817)

Fp = 0.9591

**Maximum speedup?**

According to the notes, the max speedup is equivalent to the following:

MaxSpeed = 1/(1-Fp)

Where Fp is the fraction parallel, resulting in:

MaxSpeed = 1/(1-0.9591)

MaxSpeed = 1/(0.0409)

**MaxSpeed = 24.4498**

I think I’ve done the calculations correctly, and as such have a maxspeed value of **24.4498**. To get this answer, I followed the slides on Ahmdal’s law, and plugged in my respective values.