Project #2
Numeric Integration with OpenMP Reduction

Michael Smith
Email: smitmic5@oregonstate.edu

Machine this ran on:

- I ran this on my MacBook Pro running on OS Monterey version 12.0.1:
  CPU: 2 GHz Dual-Core Intel Core i5
  Memory: 8 GB 1867 MHz LPDDR3

Predicted Volume:

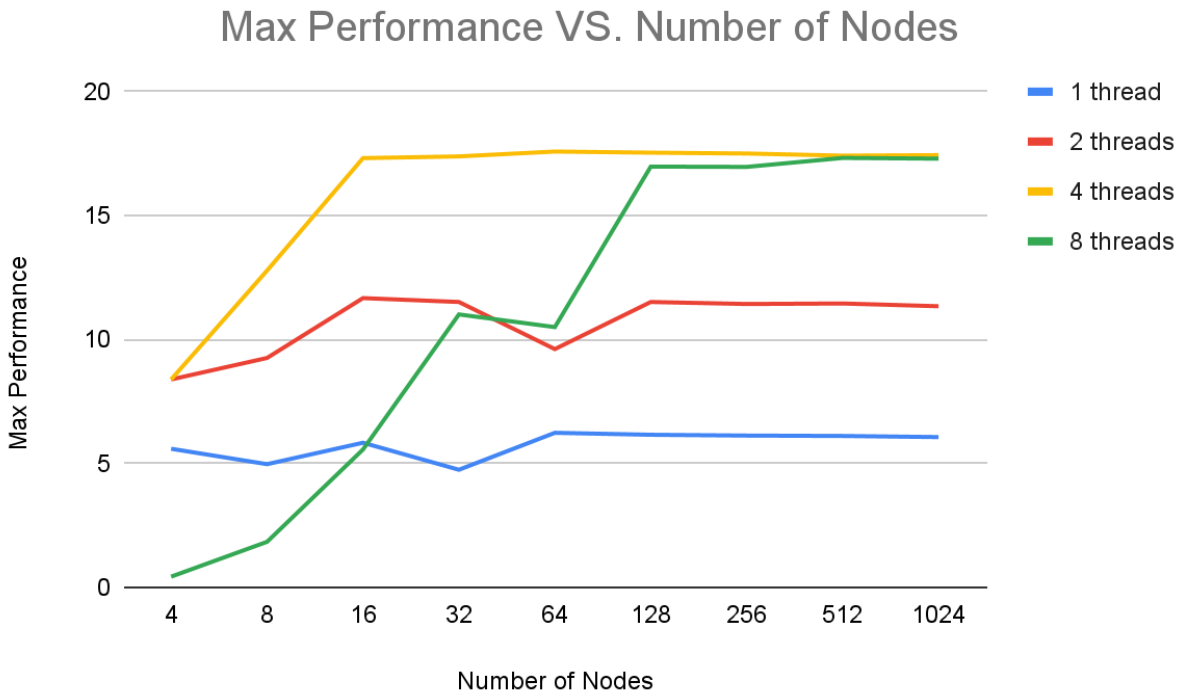
- Judging from my own calculations and research, I assumed the volume would be around 4, maybe even higher. After collecting the data, I saw that the true volume approached 3.92 as the number of threads and nodes increased. So I would say my prediction was not all that off.

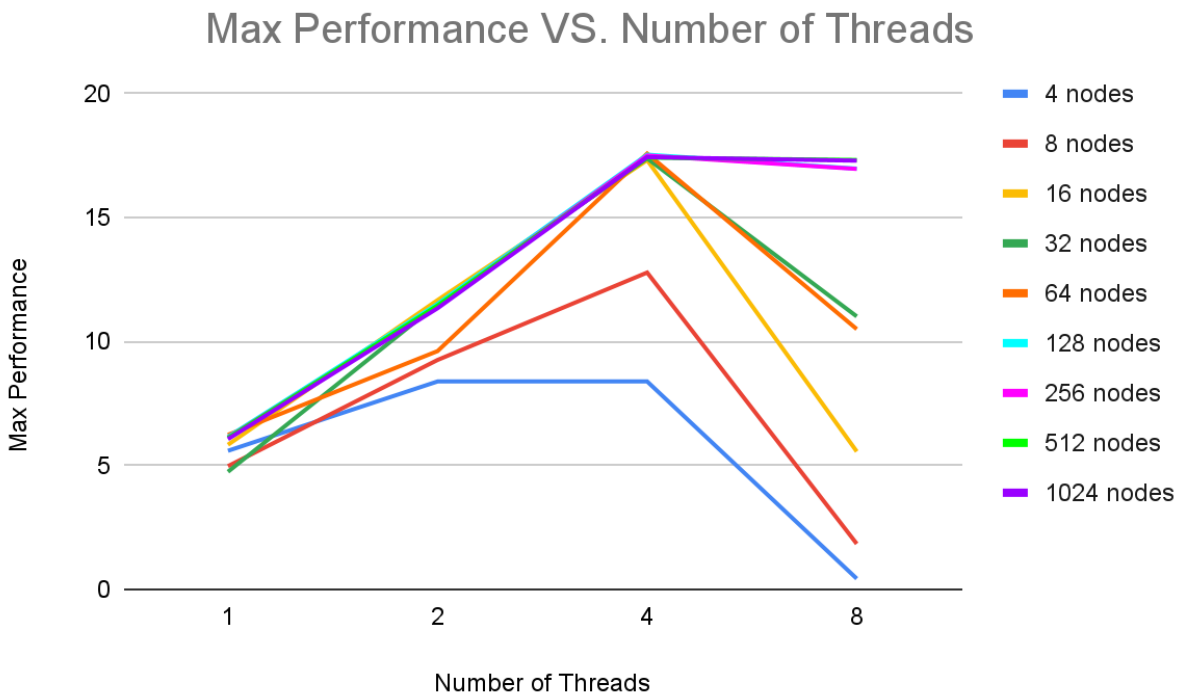
Table of Data:

- This table shows the max performance results given the number of threads and the number of nodes.

	1 thread	**2 threads**	**4 threads**	**8 threads**
4 nodes	5.592405	8.388608	8.388608	0.43296
8 nodes	4.971027	9.256395	12.78264	1.838599
16 nodes	5.835553	11.671106	17.318417	5.563429
32 nodes	4.74582	11.514658	17.388531	11.012736
64 nodes	6.235887	9.613805	17.584309	10.501143
128 nodes	6.157107	11.514658	17.539427	16.976156
256 nodes	6.124321	11.433238	17.504803	16.96463
512 nodes	6.108873	11.453843	17.418282	17.326057
1024 nodes	6.063837	11.343182	17.439348	17.297846

Graphs:


Max Performance VS. Number of Nodes
Max Performance
20
15
10
5
0
4
8
16
32
64
128
256
512
1024
Number of Nodes
1 thread
2 threads
4 threads
8 threads


Max Performance VS. Number of Threads
Max Performance
20
15
10
5
0
1
2
4
8
Number of Threads
4 nodes
8 nodes
16 nodes
32 nodes
64 nodes
128 nodes
256 nodes
512 nodes
1024 nodes

Speed Patterns and why:

- The trends I see are in the second graph, Max Performance VS. Number of Threads, and it is that the differences between the performances of different nodes increase as the number of threads increase. In other words, the lines representing the number of nodes spread as the number of threads increase. I also noticed that the highest performance tends to occur with 4 & 8 threads which are shown in both graphs. I would assume this is because with a single thread with many nodes would decrease the overall performance whereas 4 & 8 would maintain a steady one. The difference between performances I think is because having more threads than nodes hurts the overall performance, resulting in a greater performance differential.

Parallel Fraction:

- First, find the 1 to 8 Speed Up:
  SP = (Performance with 8 threads) / (Performance with 1 thread)
  SP = 17.298 / 6.064 = 2.853
  SP = 2.853
- Use the **SP** to calculate the Parallel Fraction:
  n = 8, **SP** = 2.853
  FP = (n / (n - 1)) x (1 - (1/**SP**))
  FP = (8/7) x (1 - 0.3505)
  FP = (1.14)(0.6495) = 0.7404
  FP = 0.7404

Max Speed-up:

- The max speed-up given the found **FP**:
  Max Speed-up = 1 / (1 - **FP**)
  Max Speed-up = 1 / (1 - 0.7404)
  Max Speed-up = 1 / 0.2596
  Max Speed-up = 3.852

- So the max speed up is equal to 3.852