April James

jamesap@oregonstate.edu

CS475 Parallel Programming

April 25 2021

Project 2: Numeric Integration with OpenMP Reduction

1. Machine:

Lenovo Yoga, running Windows 10. The program was ran on OSU's Flip server, with an uptime of 42 users, load average of 8.28, 8.20, 8.23.

2. SuperQuad volume using N=0.7:

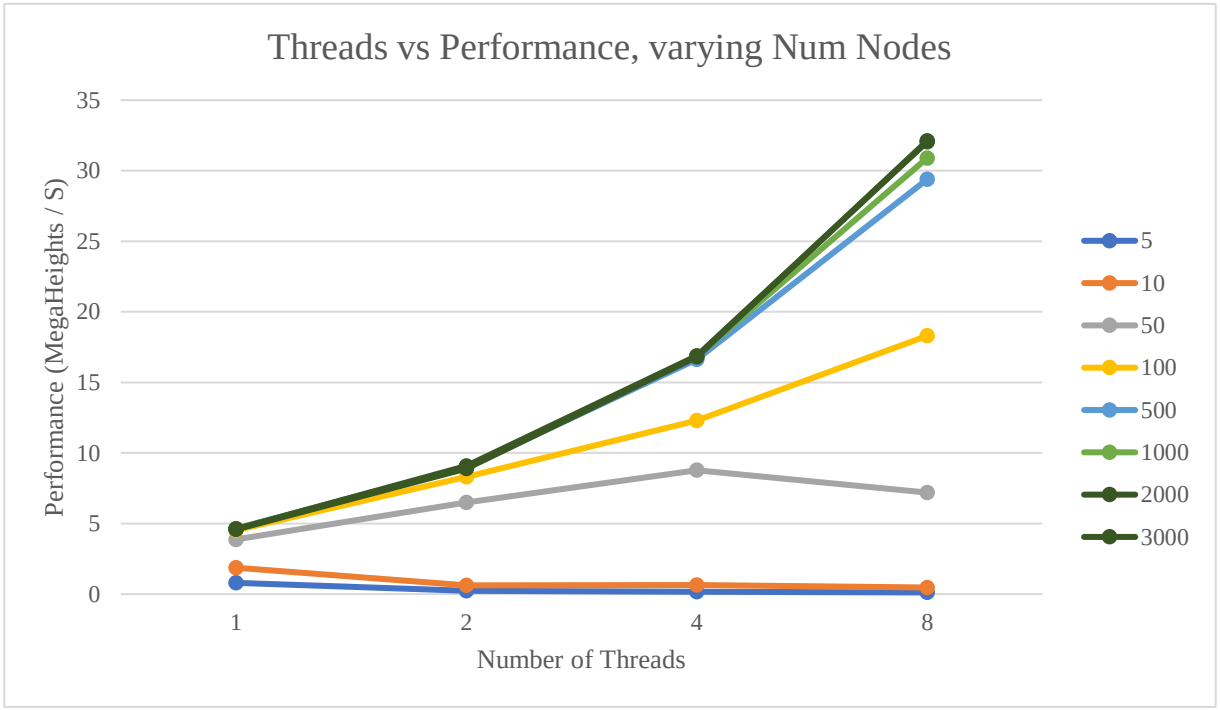
**Calculated volume: 0.4357**

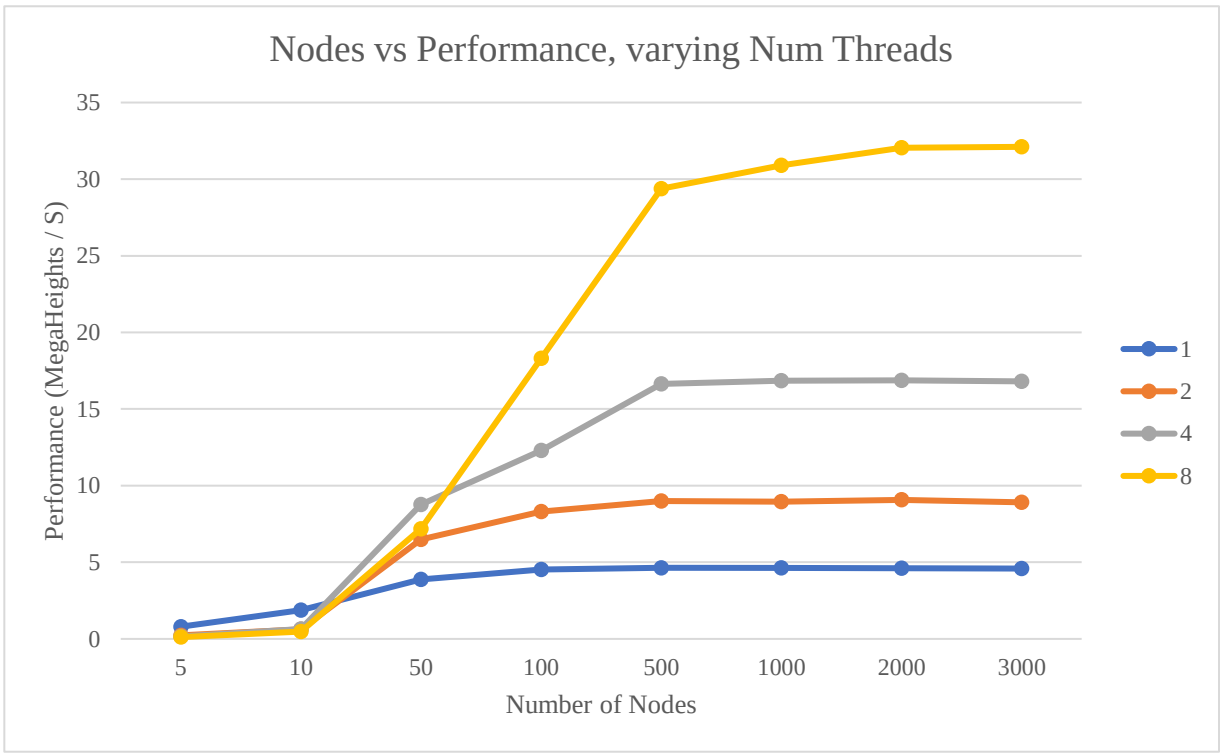
3. Performance:

Table:

**Performance (MegaHeights / second) based on Num Nodes vs Num Threads**

| | | **Number of Threads** | | | |
|---|---|---|---|---|---|
| | | **1** | **2** | **4** | **8** |
| **Number of Nodes** | **5** | 1.0104 | 1.0104 | 1.0104 | 1.0104 |
| | **10** | 0.382288 | 0.382288 | 0.382288 | 0.382288 |
| | **50** | 0.43235 | 0.43235 | 0.43235 | 0.43235 |
| | **100** | 0.434728 | 0.434728 | 0.434729 | 0.434729 |
| | **500** | 0.435711 | 0.435714 | 0.435714 | 0.435713 |
| | **1000** | 0.435743 | 0.43576 | 0.435762 | 0.43576 |
| | **2000** | 0.435611 | 0.435724 | 0.435742 | 0.43576 |
| | **3000** | 0.435137 | 0.435581 | 0.435654 | 0.435723 |


Threads vs Performance, varying Num Nodes
Performance (MegaHeights / S)
35
30
25
20
15
10
5
0
1
2
4
8
Number of Threads
5
10
50
100
500
1000
2000
3000


Nodes vs Performance, varying Num Threads
Performance (MegaHeights / S)
35
30
25
20
15
10
5
0
5
10
50
100
500
1000
2000
3000
Number of Nodes
1
2
4
8

4. Patterns in speeds:

The most obvious pattern with speed is the relationship between number of threads and performance especially when the number of nodes, or calculations, is high. When the number of threads increases, the performance increases significantly. However, the lower the number of nodes, the less the speedup is with more threads.

5. Why is this happening?

The reason why speedup does not occur as thread count increases when the number of calculations is relatively small is because there is not much work to be parallelized. Even though many threads are present to divide the parallelizable calculations, this amount of work is very small and does not decrease the overall time much since the initial overhead work must still run in series.

However, when the number of calculations is very large, the amount of work that can be parallelized increases, which causes the speedup when more threads are introduced to be much greater. This is Gustavson's Law in full effect!

6. Parallel Fraction:

$$F_p = \frac{n}{n-1} * \left(1 - \frac{1}{S}\right)$$

$$F_p = \frac{8}{8-1} * \left(1 - \frac{1}{P_n / P_1}\right)$$

$$F_p = \frac{8}{7} * \left(1 - \frac{1}{32.1072 / 4.58227}\right)$$

$$\underline{\boldsymbol{F_p = 0.979}}$$

7. Maximum speed-up *ever*…

$$\max speedup = \frac{1}{1 - F_P}$$

$$\max speedup = \frac{1}{1 - 0.979}$$

$$\underline{\mathbf{\max speedup = 49.385}}$$