CS 475

4/21/2021

Kyle Huang

[huangky@oregonstate.edu](mailto:huangky@oregonstate.edu)

Project 2

1. Tell what machine you ran this on.

My computer running Intel I9 9900k @5.1GHz / 4.8~9GHz all core maxed out. ~30% background load while running tests.

1. What do you think the actual volume is?

Based off the calculations I would say that the actual volume is about 0.436. The volumes I got were all around 0.435xxx with the exception of 100,000 nodes; likely due to ULPs.

1. 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.
   2. Performance as a function of NUMT with colored lines showing different NUMNODES values.

Performance (MegaNodes/Sec) as Threads vs # nodes

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 100 | 500 | 1000 | 2500 | 5000 | 7500 | 10000 | 25000 | 100000 |
| 1 | 5.438034 | 5.394279 | 5.403244 | 5.371156 | 5.305436 | 5.306353 | 5.347644 | 5.333066 | 5.698177 |
| 2 | 10.73192 | 10.68911 | 10.72431 | 10.56292 | 10.54962 | 10.45275 | 10.56879 | 10.56319 | 11.31031 |
| 3 | 14.97006 | 14.97975 | 15.01269 | 14.88039 | 14.75201 | 14.65993 | 14.60499 | 14.60044 | 16.75017 |
| 4 | 20.43736 | 20.43168 | 20.50037 | 19.99499 | 19.6761 | 19.82783 | 19.63161 | 19.6035 | 21.70812 |
| 5 | 24.57606 | 24.25654 | 23.91166 | 23.91219 | 23.4092 | 23.00971 | 23.13649 | 23.07017 | 26.5736 |
| 6 | 30.04808 | 29.82795 | 29.09946 | 28.07441 | 27.99735 | 27.57296 | 27.43363 | 27.41089 | 31.38373 |
| 7 | 34.15301 | 29.03702 | 32.06218 | 31.3221 | 30.34842 | 30.56793 | 30.72748 | 30.70173 | 35.36149 |
| 8 | 31.25 | 35.23211 | 36.26684 | 35.02988 | 34.32733 | 34.40307 | 33.82459 | 33.42844 | 38.71822 |
| 9 | 36.87316 | 35.84898 | 37.06436 | 37.00723 | 36.31963 | 36.11192 | 36.16385 | 35.98676 | 41.55093 |
| 10 | 38.1971 | 39.32487 | 39.86128 | 40.01721 | 39.2243 | 39.64813 | 39.61789 | 39.79267 | 44.73378 |
| 11 | 41.64931 | 45.23986 | 44.59985 | 42.33743 | 41.89708 | 41.44701 | 41.56397 | 41.73904 | 48.0719 |
| 12 | 38.24092 | 46.59745 | 44.29169 | 43.69554 | 44.16395 | 43.83052 | 44.42908 | 44.84232 | 51.07685 |
| 13 | 48.42615 | 48.77287 | 48.95266 | 47.59011 | 46.30871 | 46.02718 | 46.55138 | 46.42656 | 53.9944 |
| 14 | 53.3049 | 53.35382 | 51.58096 | 49.81429 | 49.16336 | 49.17934 | 49.64478 | 49.99028 | 56.71394 |
| 15 | 56.24297 | 56.73437 | 54.46564 | 51.51207 | 51.20081 | 51.07115 | 50.90562 | 50.95568 | 57.78553 |
| 16 | 37.72162 | 40.13614 | 43.46711 | 47.19485 | 51.72625 | 51.3822 | 51.0061 | 51.96852 | 58.72207 |

Volume as Threads vs # nodes

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 100 | 500 | 1000 | 2500 | 5000 | 7500 | 10000 | 25000 | 100000 |
| 1 | 0.434729 | 0.435714 | 0.435761 | 0.435778 | 0.435781 | 0.435782 | 0.435782 | 0.435782 | 0.001004 |
| 2 | 0.434729 | 0.435714 | 0.435761 | 0.435778 | 0.435781 | 0.435782 | 0.435782 | 0.435782 | 0.001004 |
| 3 | 0.434729 | 0.435714 | 0.435761 | 0.435778 | 0.435781 | 0.435782 | 0.435782 | 0.435782 | 0.001004 |
| 4 | 0.434729 | 0.435714 | 0.435761 | 0.435778 | 0.435781 | 0.435782 | 0.435782 | 0.435782 | 0.001004 |
| 5 | 0.434729 | 0.435714 | 0.435761 | 0.435778 | 0.435781 | 0.435782 | 0.435782 | 0.435782 | 0.001004 |
| 6 | 0.434729 | 0.435714 | 0.435761 | 0.435778 | 0.435781 | 0.435782 | 0.435782 | 0.435782 | 0.001004 |
| 7 | 0.434729 | 0.435714 | 0.435761 | 0.435778 | 0.435781 | 0.435782 | 0.435782 | 0.435782 | 0.001004 |
| 8 | 0.434729 | 0.435714 | 0.435761 | 0.435778 | 0.435781 | 0.435782 | 0.435782 | 0.435782 | 0.001004 |
| 9 | 0.434729 | 0.435714 | 0.435761 | 0.435778 | 0.435781 | 0.435782 | 0.435782 | 0.435782 | 0.001004 |
| 10 | 0.434729 | 0.435714 | 0.435761 | 0.435778 | 0.435781 | 0.435782 | 0.435782 | 0.435782 | 0.001004 |
| 11 | 0.434729 | 0.435714 | 0.435761 | 0.435778 | 0.435781 | 0.435782 | 0.435782 | 0.435782 | 0.001004 |
| 12 | 0.434729 | 0.435714 | 0.435761 | 0.435778 | 0.435781 | 0.435782 | 0.435782 | 0.435782 | 0.001004 |
| 13 | 0.434729 | 0.435714 | 0.435761 | 0.435778 | 0.435781 | 0.435782 | 0.435782 | 0.435782 | 0.001004 |
| 14 | 0.434729 | 0.435714 | 0.435761 | 0.435778 | 0.435781 | 0.435782 | 0.435782 | 0.435782 | 0.001004 |
| 15 | 0.434729 | 0.435714 | 0.435761 | 0.435778 | 0.435781 | 0.435782 | 0.435782 | 0.435782 | 0.001004 |
| 16 | 0.434729 | 0.435714 | 0.435761 | 0.435778 | 0.435781 | 0.435782 | 0.435782 | 0.435782 | 0.001004 |

1. What patterns are you seeing in the speeds?

One of the patterns that I see is that the number of nodes does not seem to affect the performance all that much. The only time the number of nodes affects the performance is when the core to node ratio is high. This pattern can also be seen in the performance vs number of threads graph as in the higher thread count, the low node count significantly reduces performance.

There is also a decrease in speedup gained from adding more threads that seems to take place around 7 threads.

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

The first pattern, static performance while increasing nodes, that was noticed would be due to the CPU’s clock speed limits. This is excluding the higher thread count where overhead would be an issue, which is another pattern. The larger thread count having poor performance on low node count would be due to the overhead in setting up more threads. The CPU is fast enough to do all of the calculations on the lower number of nodes in a short amount of time so setting up more cores and having only one thread able to write to volume at a time is shown to significantly reduce performance on low node counts.

The decrease in speedup from adding more threads could be due to the threads all writing to the same buffer. This would cause more threads to be waiting than calculating, where the sequential portion of the code starts causing noticeable speedup decrease. I would assume this would be less of an issue if there were more calculations or instructions in between writing time so that there is a lower chance of threads trying to write at the same time. Another issue that could cause some slowdown with higher cores is that my CPU was hitting a power limit. The higher node counts would make the threads that were being used to go to 100% utilization and eventually it will thermal throttle. This causes some issues since the CPU will be running slower once it thermal throttles, giving inaccurate data for the higher thread counts after the CPU decides to thermal throttle.

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

I am calculating using 100,000 nodes and a speed up from 1 to 16 cores. I am using 100,000 nodes because the data contains the least amount of “noise” (overhead, ect. Will be less of an issue).

S =

Fp=

Fp = 0.963164 ≈ 96.3%

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

Max Speedup =