Saket Roy
April 2, 2020

Machine Specifications:

Operating System: **GNU/Linux, Linux Mint 19.3 Tricia**

CPU : **Intel(R) Core(TM) i5-4200M CPU @ 2.50GHz**

Performance Results:

I modified the program to output the Average Performance as well to get a better sense of my machine's performance.

SIZE = 32768

NUMTRIES = 1000

Using 4 threads

Peak Performance = 7289.12 MegaMults/Sec

Avg Performance = 6203.95 MegaMults/Sec

Peak Performance ranged between 6800-7300 MegaMults/Sec

Avg Performance ranged between 4000-6500 MegaMults/Sec

Using 1 threads

Peak Performance = 2979.45 MegaMults/Sec

Avg Performance = 2120.53 MegaMults/Sec

Peak Performance ranged between 2800-3100 MegaMults/Sec

Avg Performance ranged between 1900-2400 MegaMults/Sec

Given the sizable difference between Average and Peak Performance for both 1 and 4 threads, I decided to rerun the program with SIZE = 65536 and NUMTRIES = 5000. This time I got more consistent results:

Using 4 threads

Peak Performance = 5787.36 MegaMults/Sec

Avg Performance = 5502.63 MegaMults/Sec

Using 1 threads
Peak Performance = 2660.81 MegaMults/Sec
Avg Performance = 2527.10 MegaMults/Sec

Speedup Fraction:

Peak Performance with 1 thread : 2660.81 MegaMults/Sec
Peak Performance with 4 threads: 5787.36 MegaMults/Sec
Speedup Fraction(S) = 5787.36/2660.81
= 2.175

As for the question of why the speedup is less than 4.0, I am assuming it is because of the overhead of creating additional threads (i.e. maintaining the state for each thread). Even when I increase the SIZE to 131702 or NUMTRIES to > 10000, I do not see any significant gain in Speedup Fraction. On the other hand, decreasing the SIZE and NUMTRIES gives me better Speedup fractions, as in the 1st case where I tested with SIZE = 32768 and NUMTRIES = 1000, I got a Speedup fraction of 2.447; regardless, my speedup fraction remains less than 3.0, so I'm assuming my machine is a factor in this as well? To test this I ran the program on a Macbook Pro (High Sierra, Intel(R) Core(TM) i5-7360U CPU @ 2.30GHz), and got the following results:

Using 4 threads
Peak Performance = 941.36 MegaMults/Sec
Avg Performance = 704.46 MegaMults/Sec

Using 1 threads
Peak Performance = 345.32 MegaMults/Sec
Avg Performance = 330.46 MegaMults/Sec

SIZE = 32768
NUMTRIES = 1000

This results in a Speedup Fraction of 2.726

Note: Using bigger values like SIZE = 65536, 131072 and NUMTRIES = 2500, 5000 did not result in noticeable difference in the Speedup Fraction.

It seems to me that machine specifications did have a factor to play although not as strongly as I had suspected. However, a key difference is that I ran the code on the Linux machine with the optimization flag (-o3) which I was unable to on the Macbook. Given that the Macbook was a borrowed machine I did not want to mess with it too much. Perhaps that might have skewed the results.

Parallel Fraction:

$$Fp = (4.0/3.0)*(1.0 - (1.0/S))$$
$$= 0.7203$$