MTCS-103P Lab Work- 10

Sieve of Eratosthenes

Regd no: 22555

Sieve of Eratosthenes is a simple algorithm to find the prime numbers up to any given limit. It is one of the most efficient ways to find small prime numbers. For a given upper limit n the algorithm works by iteratively marking the multiples of primes as composite, starting from 2. Once all multiples of 2 have been marked composite, the multiples of the following prime number, i.e 3 are marked composite. This process continues until p≤√n where p is a prime number.

Implementation

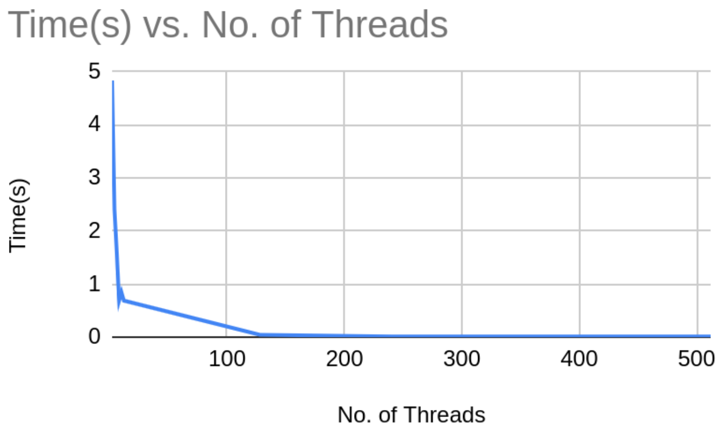
1. To find out all primes under n, generate a list of all integers from 2 to n.
2. Start with the smallest prime number, ie p = 2
3. Mark all the multiples of p which are less than n as composite. To do this, mark the value of the numbers (multiples of p) in the generated list. Do not mark p itself as composite.
4. Assign the value of p to the next prime. The next prime is the next non-zero number in the list which is greater than p.
5. Repeat the process until p≤√n

Results:

Openmp:

Size:100000000

No. of Threads	Time(s)
2	4.834
4	2.41
6	1.587
8	0.6898
10	0.8459
12	0.6898
128	0.04626
240	0.01668
512	0.01336

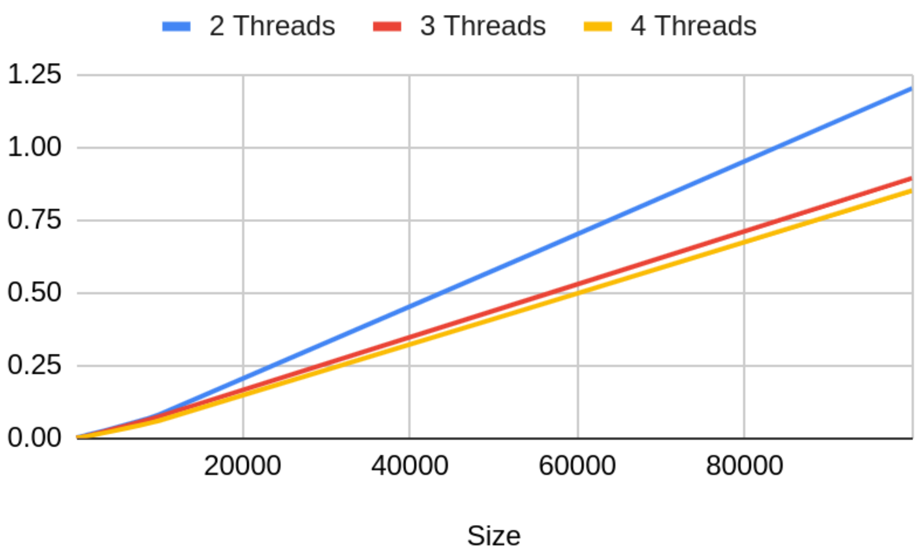


Here we find that as the number of threads was less there was a substantial decrease in the time that was taken to find the prime numbers. As the number of threads increased, the time taken remained constant. We see that towards the end of the graph.

MPI:

The parallel code using MPI was also implemented and the results were taken for different sizes for 2,3 and 4 processes and the computer contains only 4 cores. There was a significant decrease in time taken when the number of processes was increased.

		Time Taken(s)	
Size	2 Threads	3 Threads	4 Threads
10000	0.0007558	0.000519	0.000324
100000	0.007159	0.0040674	0.0038217
1000000	0.079843	0.0739492	0.059458
10000000	1.2024612	0.8937398	0.8507558


2 Threads
3 Threads
4 Threads
Size