

## OpenMP Implementation

The Mandelbrot set was computed using the **OpenMP** parallelization technique with varying numbers of threads. The serial algorithm was used as a baseline, and the following values for the number of iterations were tested:

```
iteration_values = [10, 25, 50, 100, 250, 500, 1000, 2500, 5000]
```

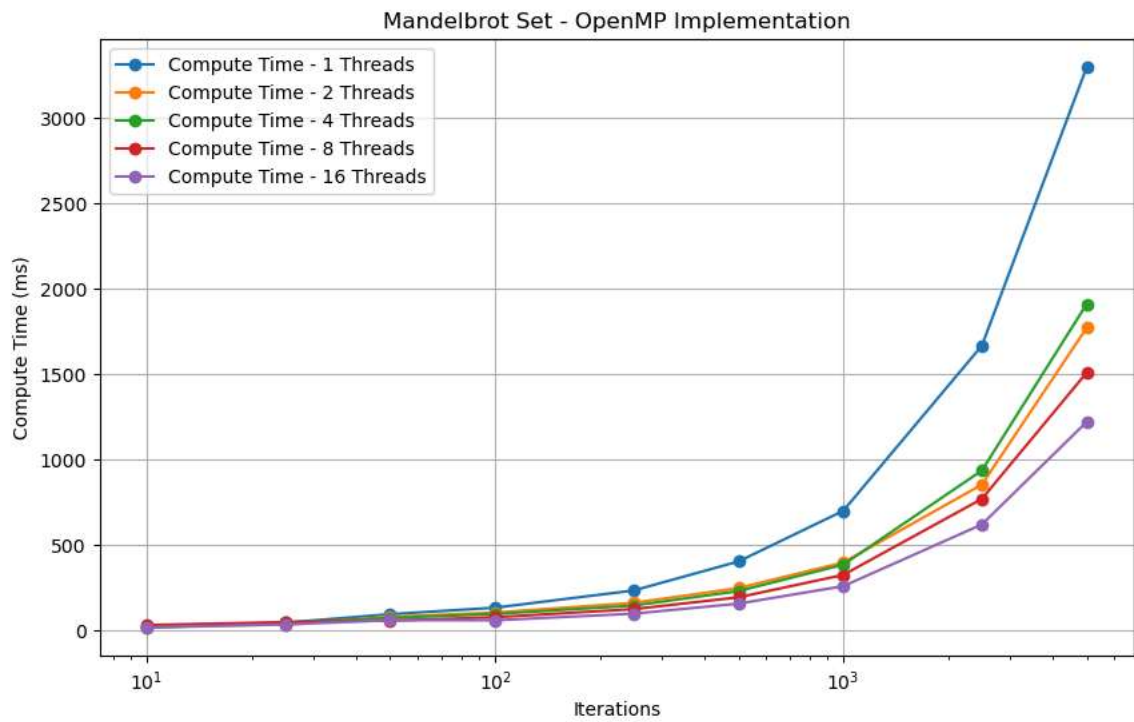
The computation was run for a total of 10 times per iteration value, and the graph represents the **average** time taken per iteration for different thread counts.

The parallel version of the algorithm was tested with the following numbers of threads:

```
num_threads = [1, 2, 4, 8, 16]
```

## Performance Metrics

For better clarity, the x-axis of the graph is logarithmic to represent the iteration values, and the y-axis is linear to represent the compute times in milliseconds.



Data:

Threads	Iterations	Compute Time (ms)	Save Time (ms)
1	10	22.90	24.40
1	25	46.20	32.30
1	50	96.20	19.20
1	100	134.10	17.10
1	250	234.90	18.70

1	500	404.90	18.90
1	1000	701.70	24.10
1	2500	1669.00	25.90
1	5000	3299.90	20.30
2	10	25.10	28.60
2	25	35.10	24.70
2	50	81.10	18.20
2	100	105.70	17.60
2	250	162.40	17.40
2	500	249.00	17.30
2	1000	399.00	19.60
2	2500	855.10	21.80
2	5000	1773.50	23.20
4	10	16.80	29.80
4	25	46.60	20.40
4	50	74.30	17.60
4	100	97.20	17.50
4	250	146.60	20.10
4	500	231.30	18.70
4	1000	386.20	16.80
4	2500	937.60	18.80
4	5000	1912.70	21.80
8	10	32.20	22.20
8	25	49.80	25.80
8	50	58.80	19.70
8	100	78.10	18.20
8	250	125.80	17.80
8	500	194.90	17.50
8	1000	326.20	19.90
8	2500	771.40	17.80
8	5000	1511.10	19.60
16	10	18.40	28.20
16	25	36.20	30.40
16	50	61.00	32.70
16	100	60.40	20.40

16	250	99.20	19.80
16	500	157.90	18.50
16	1000	260.20	23.30
16	2500	621.70	20.50
16	5000	1222.40	20.80