used technique:We used C#/.NET (?) as our programming language, and since C# does not have OpenMP we relied on Parallel.For to handle our threads.

scheduling scheme: Scheduling is handled by .NET’s task scheduler

additional finding: /

interesting remarks: /

performance comparison on a 1024 times 1024 image and following input:  
double minX = -2.0;

double minY = -1.0;

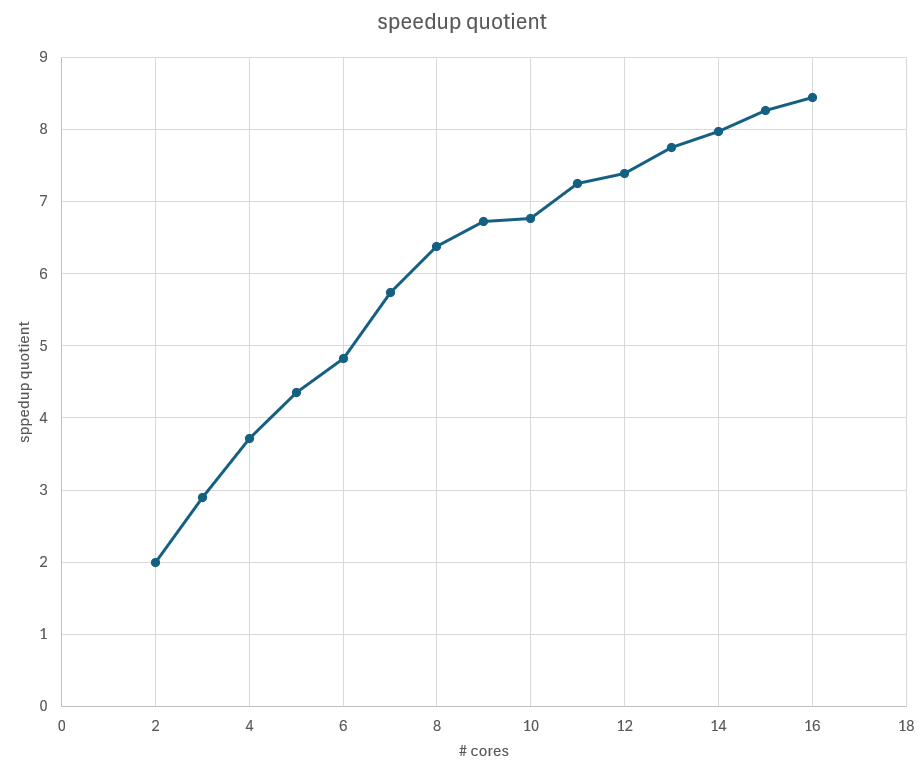
double maxX = 1.0;

double maxY = 1.0;

int maxIterations = 500;

Single Thread: 1124ms

Speedup Graph:



As you can see, the graph follows a logarithmic curve. It looks a bit wonky because we only used a single measurement. Averaging out a few more measurements would surely smooth the curve.

| # cores | time in ms |  | speedup quotient |
| --- | --- | --- | --- |
| 1 | 1124 |  |  |
| 2 | 562 |  | 2 |
| 3 | 388 |  | 2.89690722 |
| 4 | 302 |  | 3.7218543 |
| 5 | 258 |  | 4.35658915 |
| 6 | 233 |  | 4.82403433 |
| 7 | 196 |  | 5.73469388 |
| 8 | 176 |  | 6.38636364 |
| 9 | 167 |  | 6.73053892 |
| 10 | 166 |  | 6.77108434 |
| 11 | 155 |  | 7.2516129 |
| 12 | 152 |  | 7.39473684 |
| 13 | 145 |  | 7.75172414 |
| 14 | 141 |  | 7.97163121 |
| 15 | 136 |  | 8.26470588 |
| 16 | 133 |  | 8.45112782 |