Mini-Project (Part 1)

Parameters (default in code):

- pRE = 1000
- pIM = 1000
- iterations = 100
- threshold = 2
- Chunk Size Range = (1 to 200, step 10)

Performance Comparison

Computation specs:

- Intel® Core™ i5-11300H-processor

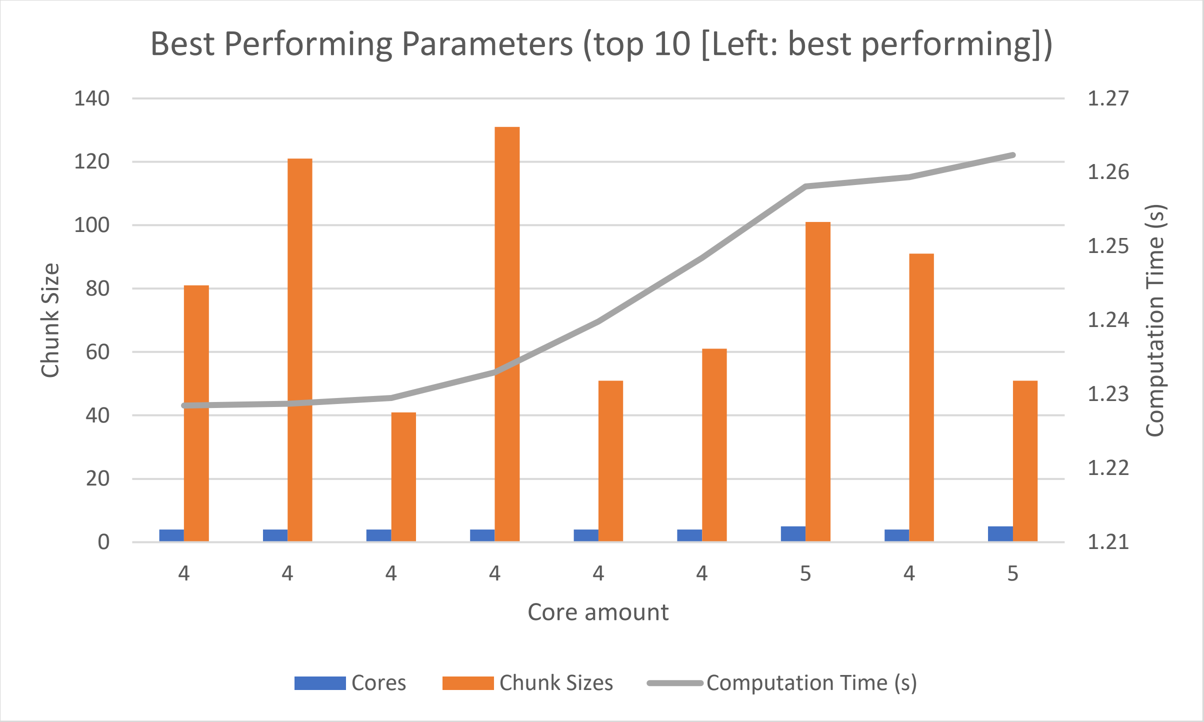
Algorithm	**Script location (root)**	**Computation Time (s)**
Naive	*mandelbrot_naive.py*	3.91s
Numpy vectorized	*mandelbrot_numpy.py*	2.12s
Numba-optimized	*mandelbrot_numba.py*	1.39s
Multiprocessing	*mandelbrot_multicore.py*	1.23s

Analysis of Multiprocessing Implementation

The 10 best performance results for different chunk sizes and processor amount

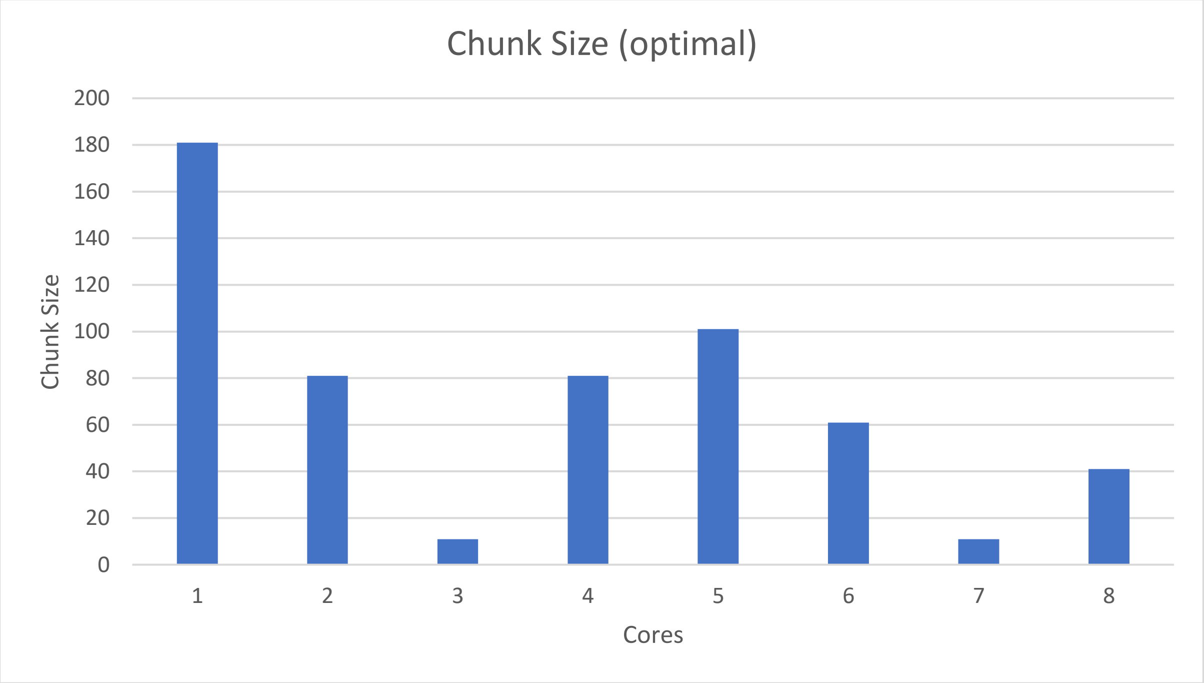
(See Computations.xlsx, Sheet: "Best Runs")

#Rank	**Number of processors**	**Chunk Size**	**Computation Time (s)**
1	4	81	1.22848
2	4	121	1.22871
3	4	41	1.2295
4	4	131	1.23295
5	4	51	1.23985
6	4	61	1.24845
7	5	101	1.2581
8	4	91	1.25935
9	5	51	1.26235
10	5	21	1.26936



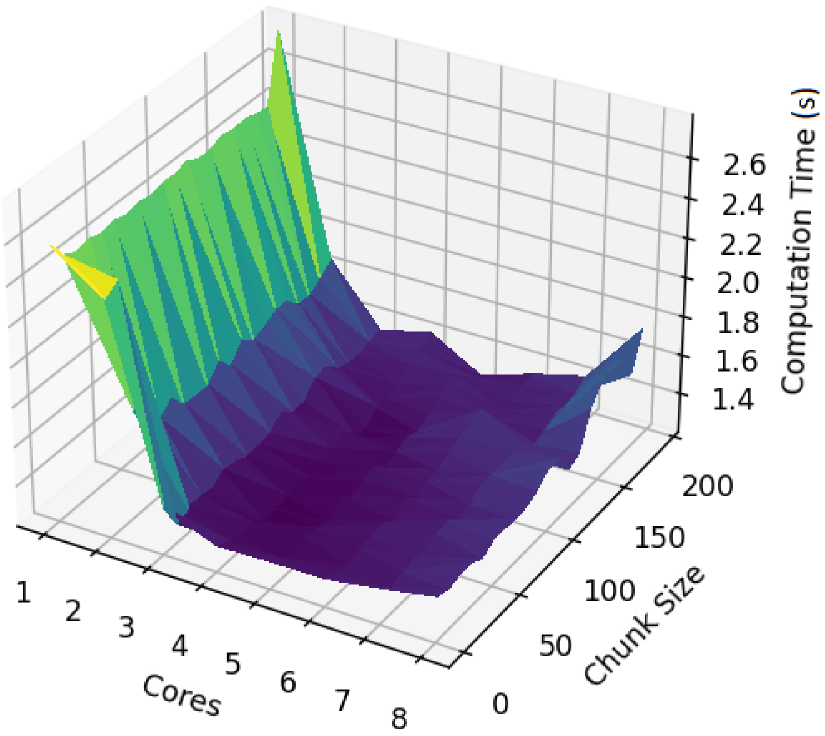
Optimal chunk size in relation to number of processes based on computation time:

(See Computations.xlsx, Sheet: "Optimal chunk size for core")

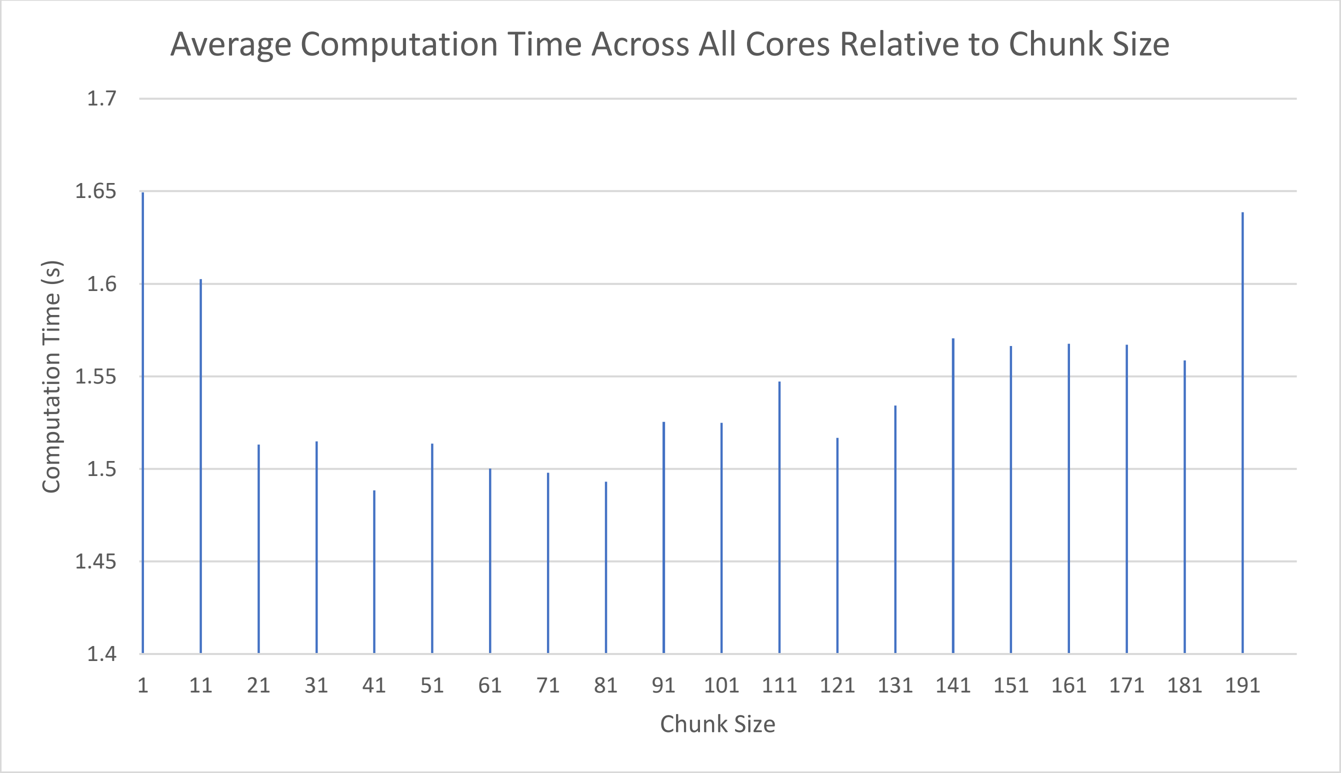


Gradient of computation time across parameters

(Graph is computed after running "mandelbrot_multicore.py")

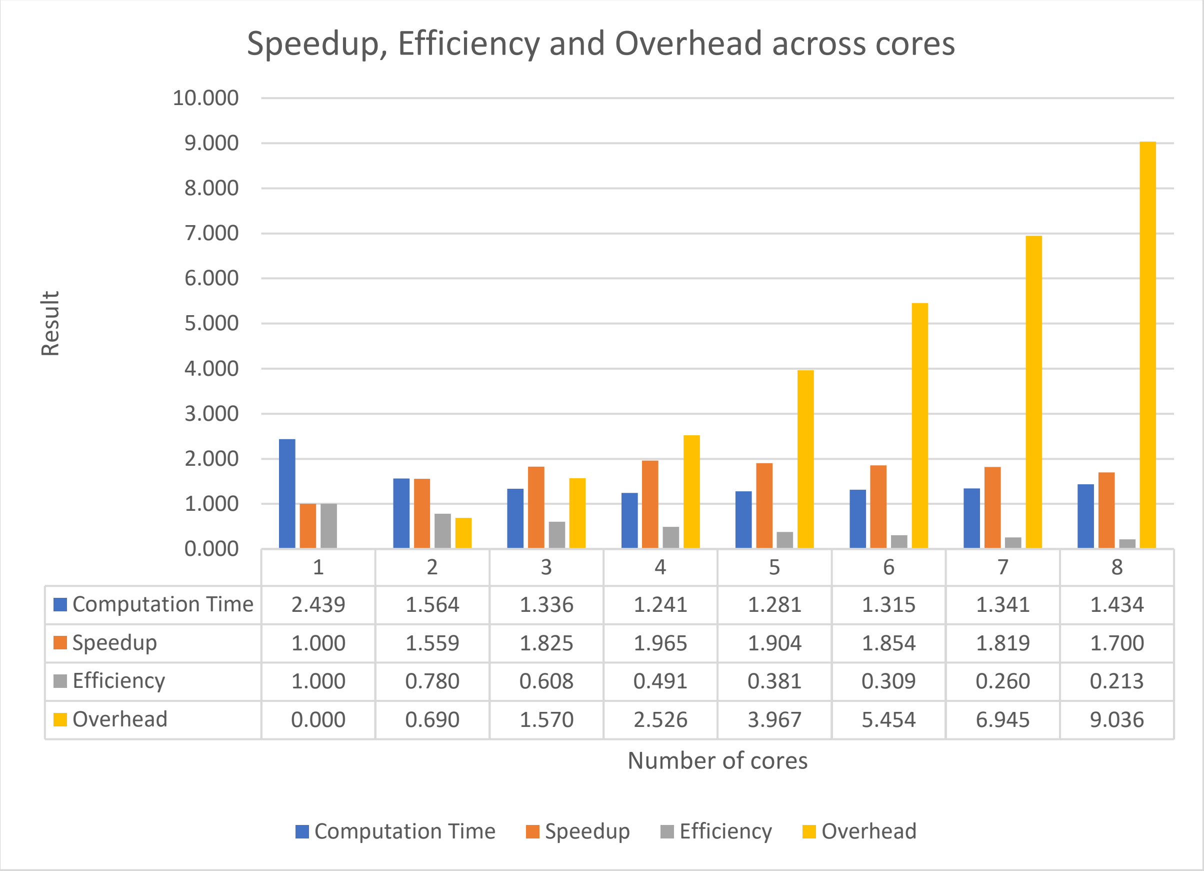


(See Computations.xlsx, Sheet: "Chunk Size")



Speedup, Efficiency and Overhead across number of cores

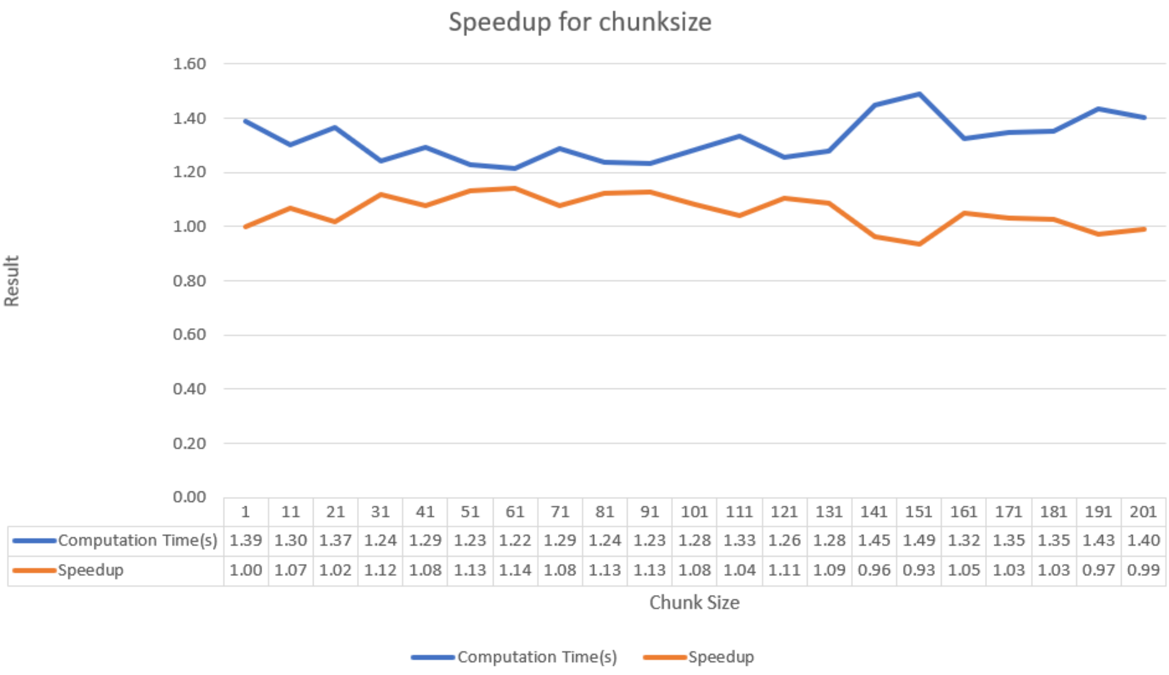
(See Computations.xlsx, Sheet: "Speedup")



	1	2	3	4	5	6	7	8
Computation Time	2.439	1.564	1.336	1.241	1.281	1.315	1.341	1.434
Speedup	1.000	1.559	1.825	1.965	1.904	1.854	1.819	1.700
Efficiency	1.000	0.780	0.608	0.491	0.381	0.309	0.260	0.213
Overhead	0.000	0.690	1.570	2.526	3.967	5.454	6.945	9.036

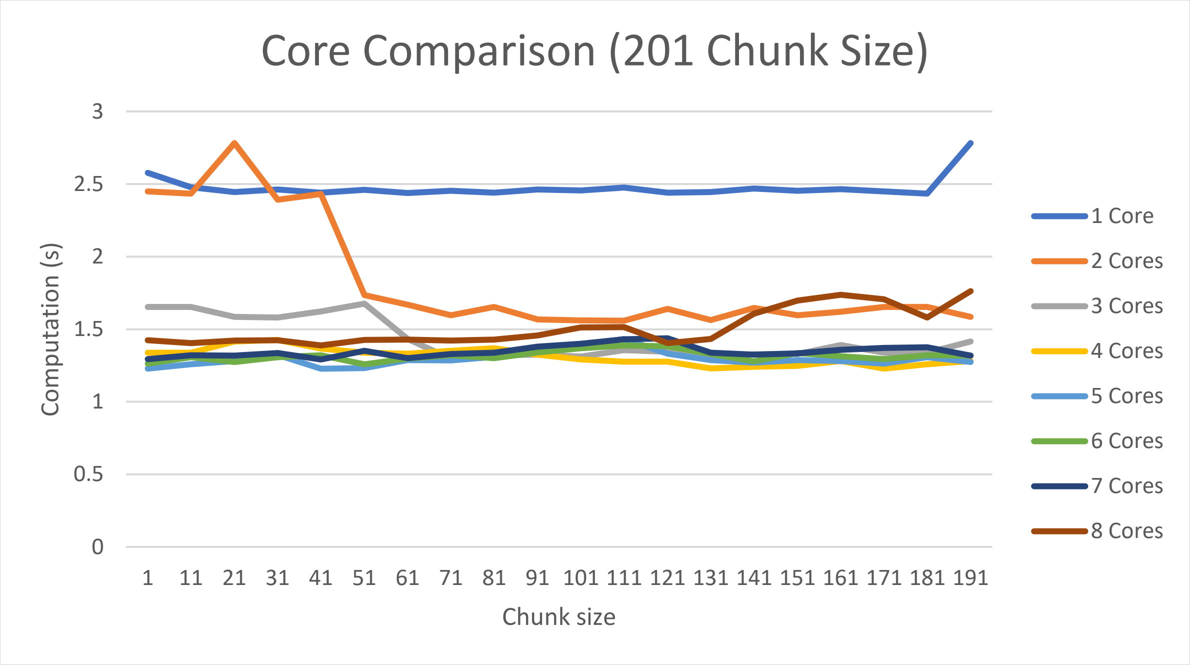
Speedup for chunksize

(See Computations.xlsx, Sheet: "Speedup")



	1	11	21	31	41	51	61	71	81	91	101	111	121	131	141	151	161	171	181	191	201
Computation Time(s)	1.39	1.30	1.37	1.24	1.29	1.23	1.22	1.29	1.24	1.23	1.28	1.33	1.26	1.28	1.45	1.49	1.32	1.35	1.35	1.43	1.40
Speedup	1.00	1.07	1.02	1.12	1.08	1.13	1.14	1.08	1.13	1.13	1.08	1.04	1.11	1.09	0.96	0.93	1.05	1.03	1.03	0.97	0.99

(See Computations.xlsx, Sheet: "Core 201 chunk size")



Output from running algorithm