Bruna Ribeiro
118171816

Avaliação

Teremos 500, 1000 e 2000 como dimensões e usaremos 1 e 2 threads. Cada caso foi testado 3 vezes e seu resultado foi anotado em *segundos*.

Hardware da máquina: 4 processadores

1° Teste

(a) Sequencial

	1 thread	2 threads
500	0.617253	0.612793
1000	7.345931	6.521201
2000	97.639966	115.955517

(b) Concorrente

	1 thread	2 threads
500	0.781654	0.426747
1000	8.560861	3.824275
2000	136.025206	43.903424

2° Teste

(a) Sequencial

	1 thread	2 threads
500	0.615516	0.653586
1000	6.537510	7.241409
2000	86.950842	69.565722

(b) Concorrente

	1 thread	2 threads
500	0.747978	0.378141
1000	7.421802	4.106353
2000	117.799850	39.889324

3° Teste

(a) Sequencial

	1 thread	2 threads
500	0.618230	0.613924
1000	5.557442	5.303086
2000	73.529591	120.614664

(b) Concorrente

	1 thread	2 threads
500	0.749654	0.378582

1000	6.533067	3.184179
2000	83.055046	45.665720

Comentários

Dá para ver que os valores sequenciais se mantêm bem parecidos mesmo ao mudar a quantidade de thread, enquanto os valores concorrentes caem bastante. Isso fica mais evidente quando comparamos o valor anterior (1 thread) com quando duas são usadas.

Quando apenas uma thread é usada, os valores entre as funções concorrentes e sequenciais são bem parecidos, e às vezes o da função sequencial é ainda menor.

Menor tempo

- 500 (1 thread): 0.615516s, sequencial
- 500 (2 threads): 0.378582s, concorrente
- 1000 (1 thread): 5.557442s, sequencial
- 1000 (2 threads): 3.184179s, concorrente
- 2000 (1 thread): 73.529591, sequencial
- 2000 (2 threads): 39.889324, concorrente