

Apresentação dos resultados

Nomes: Juliano Strelow Buss e Lucas Alviene Pereira

- **SO:** Windows 11
- **Quantidade de Memória:** 16GB
- **Cores Físicos:** 4
- **Cores Lógicos:** 8
- **GCC:** 6.3.0
- **Tamanho do Cache**
 - **L1:** 256kb
 - **L2:** 1MB
 - **L3:** 8MB

Não conseguimos adaptar o código do primeiro trabalho para o MPI, então buscamos outro algoritmo para adaptar ao MPI.

Começamos os testes com o OPENMP e MPI ativados e realizamos os teste de **N = {8, 9, 10}** para cada tempo de execução paralela **t= {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12}**.

Tempo Paralelo			
t	N		
	8	9	10
1	2.5s	1.3min	30min
2	1.3s	32s	16min
3	1.0s	23s	10min
4	0.85s	19s	8min
5	0.71s	17s	7.76min
6	0.71s	15s	7min
7	0.60s	14s	6.43min
8	0.59s	15s	6.16min
9	0.68s	14.70s	6.18min
10	0.64s	14.49s	6.3min
11	0.63s	14.16s	6.3min
12	0.59s	14s	6.5min

Primeiro vamos calcular o tempo sequencial para cada **N** com o `-fopenmp` desativado e sem o MPI equivalente a **t = 0**.

Tempo Sequencial			
t	N		
	8	9	10
0	1,9s	48.45s	23,86min

Para calcular o speedup, iremos dividir o t_0 com t_n

Obs: Usamos apenas o conjunto $N = \{8,9,10\}$, pois o algoritmo selecionado é exponencial e vai demorar muito tempo para realizar teste com $N > 10$

SpeedUp			
t	N		
	8	9	10
1	0.76	37.2 segundos	0.8
2	1.47	1.52	1.5
3	1.9	2.11	2.39
4	2.24	2.55	2.99
5	2.68	2.85	3.08
6	2.68	3.23	3.41
7	3.17	3.47	3.72
8	3.23	3.23	3.88
9	2.8	3.3	3.87
10	2.97	3.35	3.79
11	3.02	3.43	3.79
12	3.23	3.47	3.68