

N Rainhas

Para realizar o trabalho, foi implementado um algoritmo simples que explora todas as alternativas utilizando backtracking em C++20. Esse algoritmo foi invocado de duas formas:

- Sequencialmente;
- Utilizando OMP com N# trabalhadores = N# Rainhas.

Assim como esperado, o algoritmo sequencial executa mais rápido para quantidade pequena de rainhas, mas quando o tabuleiro se aproxima de um tamanho real (8x8) o algoritmo paralelo rapidamente torna-se mais viável, visto que o overhead de alocação de threads e divisão de tarefas torna-se negligenciável em comparação ao tempo de execução do algoritmo.



N# Execuções

Por padrão foi calculada a média de 100 execuções de cada algoritmo para cada N# Rainhas, visando diminuir qualquer viés ocorrido por interação com SO.

Tempos de execução

Os tempos, foram gravados usando um *81G3 Lenovo ideapad 320-15IKB* rodando Arch(kernel 6.2.6-zen1-1-zen) e compilador g++ 12.2.1 20230201.

Tempos de execucao					
N# Rainhas	Sequencial	OpenMP	Sequencial - OpenMP	Aceleração	Ganhador
1	0.000122ms	0.011523ms	0.011401ms	~0.011x	Sequencial
2	0.000315ms	0.116953ms	0.116638ms	~0.003x	Sequencial
3	0.000768ms	0.010283ms	0.009515ms	~0.075x	Sequencial
4	0.002424ms	0.011223ms	0.008799ms	~0.216x	Sequencial
5	0.009080ms	0.022677ms	0.013597ms	~0.400x	Sequencial
6	0.037944ms	0.111059ms	0.073115ms	~0.342x	Sequencial
7	0.164667ms	0.198120ms	0.033452ms	~0.831x	Sequencial
8	0.610865ms	0.557916ms	-0.052940ms	~1.095x	OpenMP
9	2.638783ms	0.753066ms	-1.885717ms	~3.504x	OpenMP
10	12.372526ms	3.343121ms	-9.029405ms	~3.701x	OpenMP
11	68.963979ms	22.088663ms	-46.875316ms	~3.122x	OpenMP
12	356.092198ms	95.309948ms	-260.782250ms	~3.736x	OpenMP

N# Rainhas	Sequencial	OpenMP	Sequencial - OpenMP	Aceleração	Ganhador
13	2269.950663ms	554.690446ms	-1715.260217ms	~4.092x	OpenMP
14	27664.419457ms	4283.618907ms	-23380.800550ms	~6.458x	OpenMP

Como rodar o projeto

Basta rodar `sh run.sh` na pasta raiz do projeto. Para mudar os parâmetros de execução veja o `main()`.

Flags

É possível controlar qual compilador será utilizado passando como argumento para o script, por exemplo:

```
sh run.sh clang++-13
```