

Relatório Trabalho de IPPD - 2022/2

N Rainhas em C

Integrantes: Pedro Thiago Roque, Ricardo Ferreira Padilha, Vitor Costa Gosmão, Yuri Borges Sena

Especificação do hardware para testes:

Processador: AMD Ryzen 5 1600, 6 cores, 12 threads, 3.2 GHz

RAM: 16GB, 2666MHz

Cache L1: 576 KB, **Cache L2:** 3 MB, **Cache L3:** 16 MB

Versão do compilador: 8.1.0

SO: Windows 10

Passo a passo:

- compilar com: "gcc N-Rainhas-OMP.c -fopenmp"
- executar com: `./"nomeDoArquivo".exe "tamanhoDoTabuleiro=numDeRainhas" "numeroDeThreads"`

O código C fornecido implementa uma solução para o problema das N-Rainhas utilizando OpenMP para executar em paralelo. O problema das N-Rainhas é um problema clássico de colocar N rainhas em um tabuleiro NxN sem que elas se ataquem, ou seja, que nenhuma rainha esteja na mesma linha, coluna ou diagonal que outra rainha.

Começamos definindo o tamanho do tabuleiro (n) e o número de threads a serem criadas, que podem ser passados como argumentos na linha de comando. Caso não sejam especificados, o programa utiliza valores padrão (8 para o tamanho do tabuleiro e 30 para o número de threads).

O programa então define a variável "max_iter", que representa o número máximo de iterações a serem executadas na busca por soluções. Esse valor é determinado pelo tamanho do tabuleiro, elevado à potência do número de rainhas (n^n). Essa variável é utilizada na iteração principal do programa, em que todas as possibilidades de posicionamento das rainhas são testadas.

Em seguida, o programa utiliza a função "omp_set_num_threads" para definir o número de threads que serão utilizadas na seção paralela. O laço principal do programa é então executado com uma diretiva OpenMP "#pragma omp parallel for", que distribui as iterações entre as threads, permitindo a execução paralela.

Dentro do laço principal, cada iteração é responsável por testar uma possível configuração do tabuleiro. A configuração é representada por um vetor "queen_rows", em que cada elemento representa a linha em que uma rainha está posicionada. O vetor é inicializado a partir de um código inteiro gerado a partir do número da iteração.

Em seguida, a função “verificaSolucao” é chamada para verificar se a configuração atual do tabuleiro é válida, ou seja, se nenhuma rainha está na mesma linha, coluna ou diagonal de outra rainha. Se a configuração for válida, o número de soluções encontradas é atualizado com a diretiva “#pragma omp atomic”, que garante a sincronização da atualização do número de soluções entre as threads.

Por fim, o programa imprime o tempo de execução e o número total de soluções encontradas. O código também possui um trecho de código comentado que permite imprimir na tela as soluções encontradas no formato de tabuleiro. Esse trecho de código pode ser ativado removendo os comentários do código.

Resultados:

Rainhas	Threads	Tempo de Execução (s)	Soluções Encontradas	Speedup: $S(t=0)/S(t)$
5	0	0	10	-
7	0	0.0780001	40	-
9	0	47.728	352	-
5	2	0.000999928	10	-
7	2	0.430002	40	0.18139
9	2	25.368	352	1.88
5	4	0.00100017	10	-
7	4	0.0189998	40	4.10531

9	4	14.052	352	3.39652
5	8	0.000999928	10	-
7	8	0.016	40	4.8750
9	8	9.114	352	5.2367