

Sudoku Multi-thread

Cristina Freitas Bazzano

RA: 135368

Flávio Altinier Maximiano da Silva

RA: 135749

Guilherme Bueno Andrade

RA: 135939

Verificação

Código disponível no arquivo sudoku-v.c. Recebe na entrada um jogo de sudoku completo e verifica se está resolvido corretamente. Caso contrário, aponta todos os erros no tabuleiro.

Usa 27 threads para percorrer todo o tabuleiro: uma para cada linha, coluna e caixa 3x3.

Alternativamente, no repositório pode ser encontrado outro arquivo, sudoku-v3.c. Nessa versão do verificador, são usadas apenas 3 threads: uma percorre todas as linhas, a outra todas as colunas e a outra todas as caixas.

É notório que a abordagem com 27 threads deve ser mais eficiente. Por exemplo, se há um erro na última linha: enquanto a abordagem de 3 threads precisaria percorrer todas as linhas para encontrar o erro, no modelo escolhido essa linha tem uma thread “exclusiva”, que deve detectar o erro mais rapidamente.

Dica

Código disponível no arquivo sudoku-d.c. Recebe na entrada um jogo de sudoku incompleto e verifica a ocorrência de cada número nas linhas, colunas e quadrados do tabuleiro. A partir dessa análise de ocorrências, gera as dicas de preenchimento para cada posição vazia do sudoku.

Usa 3 threads para percorrer todo o tabuleiro: uma para cada conjunto de linhas, colunas e caixas 3x3.

Sabemos que a abordagem com 27 threads seria mais eficiente, pois existiria uma thread para cada linha, coluna e quadrado, mas quando fizemos esse código ainda não tínhamos aprendido um método Synchronize que impediria a ocorrência de acesso simultâneo, como valores lidos incorretamente ou violação de acesso. Portanto optamos pela solução com 3 threads para impedir a ocorrência desse erro.

Completando o Diagrama

Código disponível no arquivo sudoku-c.c. Para se obter a solução é preciso escrever a matriz incompleta representando o jogo em um arquivo de texto e salvá-lo no mesmo diretório com o nome “sudoku-c.in”. Após isto, basta executar o programa e ele imprimirá a solução na tela caso a matriz seja válida, ou uma mensagem informando a invalidade caso contrário.

Tendo em vista que em certas matrizes do sudoku é muito mais rápida a obtenção da

resposta partindo de certas posições, a aplicação utiliza n threads, com n = número de células não preenchidas na matriz. Cada uma destas realiza uma busca pela resposta com a utilização de um algoritmo de backtracking iniciado a partir de cada uma das n posições vazias.

Embora esta abordagem não tenha superado nos nossos testes a implementação com uma única thread, ela soluciona os diagramas em um tempo relativamente baixo e acreditamos que se houvessem n processadores, possibilitando a efetiva execução simultânea de todas as threads, o método apresentaria um desempenho muito satisfatório.