

PROJETO FINAL CD

2023/2024

Realizado por:

- Gonalo Lopes n°107572
- Leonardo Fiuza n°97772



RESOLUÇÃO DO SUDOKU: ABORDAGEM BASEADA EM COMBINAÇÕES

1. Função line_combinations

- Passos:

- 1-Percorrer o Sudoku

- 2-Identificar Valores fixos

- 3-Gerar Permutações

- 4-Inserir Valores nas Posições Originais

- 5-Verificar Validade através de um mapa onde:

- A chave é a linha.
 - O valor é outro mapa onde:
 - A chave é a coluna.
 - O valor é o número da permutação.

- 6-Agrupar e Retornar Linhas

2. Função combine_lines

- Passos:

- 1-Receber Array e Índice

- 2-Aplicar Filtro

- 3-Distribuir Trabalho

- 4-Distribuição de Pesquisa: A abordagem distribui a pesquisa a partir de uma única linha, dividindo automaticamente as possibilidades das combinações dessa linha.

Conclusão:

Ao dividir as possibilidades de uma única linha e distribuí-las entre vários nós, nossa abordagem otimiza a resolução do sudoku, permitindo uma pesquisa mais eficiente e distribuída das combinações válidas.

COMPONENTES DO PROJETO

Classe P2PNode:

- Gerencia as operações da rede P2P.
- Lida com a passagem de mensagens entre nós.
- Gerencia o processo de resolução distribuindo o trabalho entre os nós

Classe SudokuSolver:

- Implementa a lógica de resolução de Sudoku.
- Gera combinações de linhas possíveis para soluções de Sudoku.
- Valida potenciais soluções de Sudoku.

Classe SudokuRequestHandler:

- Lida com solicitações HTTP para resolver Sudoku e recuperar estatísticas.
- Faz a interface entre solicitações HTTP e a rede P2P.

O nosso script usa no P2PNode o `check()` da função do professor para verificar se o sudoku final é válido.

PROTOCOLO DE MENSAGENS

Usamos sockets UDP. Os nós P2P comunicam-se usando vários tipos de mensagens para coordenar a resolução do quebra-cabeça de Sudoku. Cada tipo de mensagem tem um propósito específico:

Join:

- Exemplo: {"type": "join", "address": "localhost:8001"}
- Quando um nó inicia e precisa se juntar à rede envia uma mensagem join para o nó principal e o nó principal trata de o adicionar à lista de peers, onde address é o address do nó principal.
- Temos um array que tem todos os peers(all_peers) e as suas conexões, quando o nó principal recebe um join, atualiza esse dicionário e adiciona o novo nó ao all_stats que tem todas as estatísticas de todos os nós e, por fim, envia tudo aos restantes nós da rede com a seguinte mensagem 1 no caso de nenhum sudoku estar a ser resolvido e 2 no caso contrário:
 1. msg = {"type": "new_peer", "all_peers": self.all_peers, "all_range_index": self.all_range_index, "solved": self.solved, "new_peer": new_peer, "sudoku": self.sudoku, 'all_stats': self.all_stats}
 2. msg = {"type": "new_peer", "all_peers": self.all_peers, "solved": self.solved, 'all_stats': self.all_stats}

Solution:

- Exemplo: {"type": "solution", "sudoku": [...], "from": "localhost:8001"}
- Quando um nó encontra uma solução, caso seja valida(verificamos usando o check() do professor), envia o sudoku aos restantes. Cada nó recebe esta mensagem, incrementa o valor do all_stats["all"]["solved"] e envia para os restantes as validations que fez.
- Por fim o nó que recebeu o pedido HTTP retorna o sudoku

update_range_number:

- Exemplo: {"type": "update_range_number", "from": "localhost:8001", "is_new_peer": true}
- Usado quando um nó precisa de um novo intervalo para processar e para remover o índice da lista que vai processar e enviar para os restantes a nova lista all_ranges_indexes(lista com todos os índices, em cada índice está um conjunto de combinações para uma determinada linha que depois vão ser combinadas com as combinações das restantes linhas.

PROTOCOLO DE MENSAGENS

Disconnect:

- Exemplo: {"type": "disconnect", "address": "localhost:8001", "all_range_index": self.all_range_index, "validations": self.solver.validations}
- Quando um nó se desconecta, caso estivesse a calcular combinações para certo índice do all_range_indexes, este nó antes de se desconectar volta a adicionar esse índice à lista para depois voltar a ser processada. Este também envia as validações que fez aos restantes nós.

Stats :

- {"type": "stats", "validations": 100, "address": "localhost:8001"}
- Após um nó ter recebido uma solução ou quando um nó se desconecta. As estatísticas deste nó são atualizadas em todos os nós.

Solve :

- {"type": "solve", "sudoku": [...], "range_number": 0, "next_line_index": 1}
- Serve para iniciar o processo de resolução do Sudoku, em que o range_number é o índice do intervalo de combinações da linha que vai combinar com as restantes e next_line_index é o próximo range_number.