

Overview

Neste trabalho foi implementado um sistema distribuído capaz de resolver puzzles de sudoku com recurso a trabalho colaborativo entre diferentes nós, permitindo chegar a uma solução de um sudoku de uma forma eficiente, simples e escalável.

Para facilitar a comunicação e colaboração entre os nós na rede, é utilizado o protocolo UDP (User Datagram Protocol), que permite a troca rápida e eficiente de dados entre os diferentes nós, sem a necessidade de um servidor centralizado. Esta escolha baseia-se na alta eficiência e baixa sobrecarga do UDP, além de sua capacidade de proporcionar uma comunicação direta e sem conexão entre os nós. Isso simplifica a arquitetura e acelera a troca de informações, essenciais para garantir uma resposta rápida na resolução de quebra-cabeças de Sudoku.

Para além disto, é usada uma abordagem P2P não centralizada, que permite a cada nó comunicar com os outros diretamente sem a necessidade de haver um intermediário nessa comunicação, o que reduz a sobrecarga da comunicação.

GET */stats*

Devolve as estatísticas de resolução e validação de sudoku de todos os nós da rede atual.

GET */network*

Devolve o endereço de todos os nós que estão atualmente ligados na rede ativa.

Post */solve*

Efetua um pedido ao servidor HTTP de solução de um puzzle sudoku. Este efetua à divisão de tarefas, enviando-as para os nós conectados á rede. Quando recebe uma resposta de todos os nós, combina e valida os resultados recebidos. Esta sequência de eventos repete-se até que o resultado seja válido.

Modo de comunicação:

A comunicação entre os nós da rede é efetuada toda no formato Json. Contudo, diferem na informação que contêm:

- { "type" : "JOIN" , "address" : "<endereço>" } : enviada na criação de um novo nó, com destino ao servidor;
- { "type" : "LEAVE" , "address" : "<endereço>" } : enviada de um nó para o servidor quando este é interrompido;
- { "type" : "TASK" , "task_id" : "<INT>", "cell" : "[<row>, <column>]", "sudoku" : <puzzle> } : enviada pelo servidor para cada nó, que antes de a enviar, identificou no puzzle todas as células vazias, efetuando uma distribuição destas pelos nós da rede.
- { "type": "RESULT", "task_id": 1, "cell": [<row>, <column>], "possible_numbers": [<num1>, <num2>, ...] } : enviada por cada nó para o servidor após chegar à conclusão de todos os números possíveis que possam preencher determinada célula.
- { "type": "STATS" } : enviada por cada nó para o servidor quando é requisitada a estatística de resolução através do comando **/stats**;

- { "type": "NETWORK" }: enviada por cada nó para o servidor quando é requisitada a informação da rede através do comando **/network**;