



Peer-to-Peer Sudoku Solver

Computação Distribuída 2023/2024

Giovanni Santos 115137

Edivaldo Gustavo Castro Luís Bonfim 110124

O trabalho tem como objectivo construir uma rede p2p para a resolução de puzzles sudoku, com o objetivo de descobrir a solução no menor tempo possível e distribuir a carga de trabalho entre os vários nós da rede.

Os nós comunicam-se em json, e trocam as seguintes mensagens:

join:

- bindPoint: ip e o porto do nó que se conectou
- reply: enviar a lista de conexões; - ip: ip

join_reply:

- bindPoints: lista de endereços a qual deve se ligar
- ip: ip do nó

askToSolve: Avisar um nó que há tarefas

agToSolve: askToSolve reply, concordou em resolver

network: pedir para enviar os dados de conexão

update_network: dados atualizados da rede do peer

solve:

- sudoku: o subpuzzle que deve ser resolvido
- sudokuld: Id do puzzle

solution:

- sudoku: puzzle, resolvido ou não
- sudokuld: ID do puzzle que resolveu
- solução: se conseguiu resolver ou não (True/False)

stop: - sudokuld: ID do puzzle que deve parar de resolver

keep_alive:

- IP: IP do nó que enviou o ping
- status: status do remetente

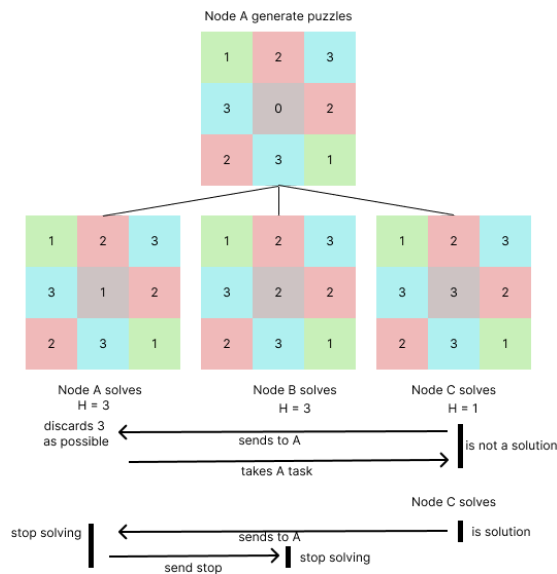
keep_alive_reply: confirm this node is alive

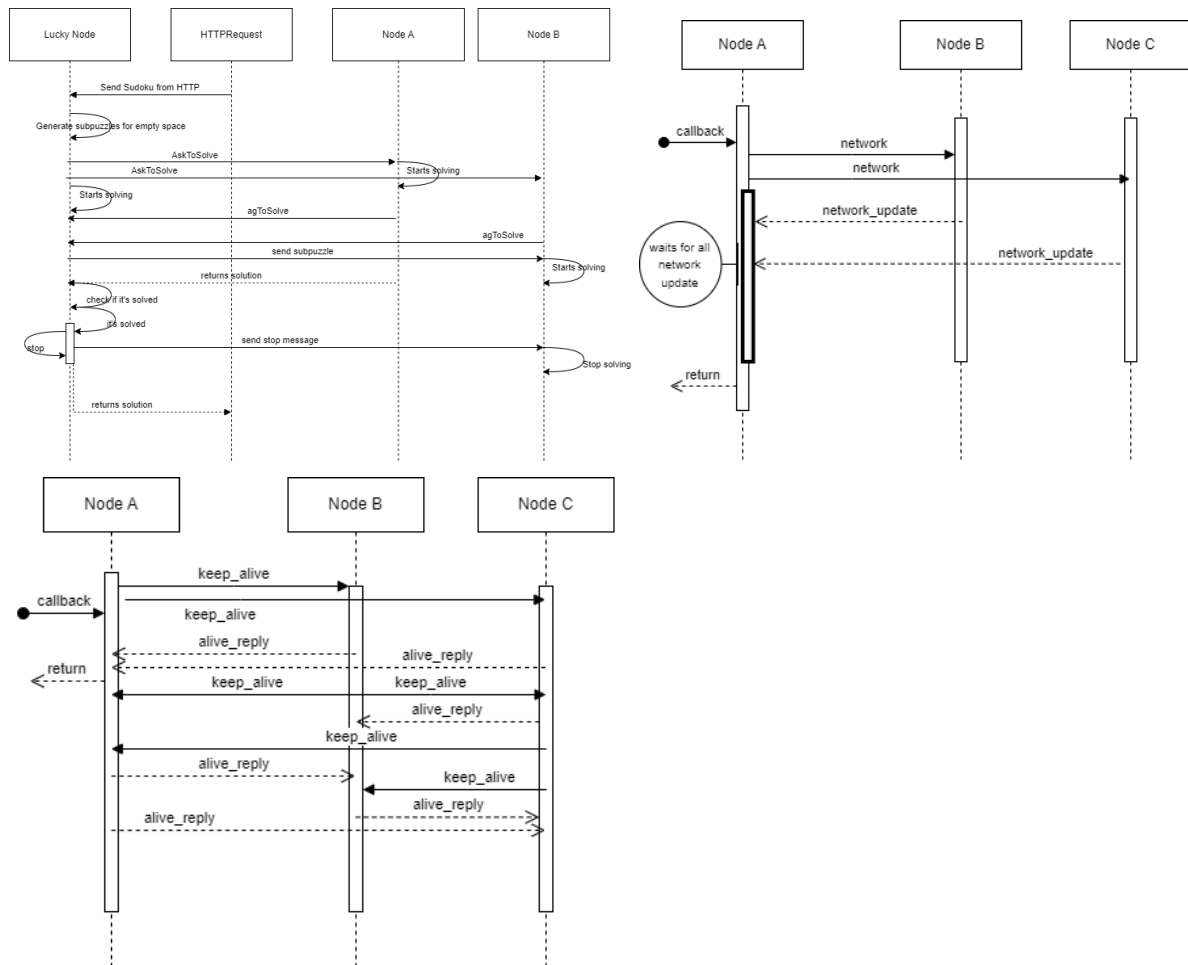
Arquitetura de Redes:

- Neste trabalho foi implementado uma arquitetura descentralizada P2P, onde todos os nós estão conectados entre si e atuam como servidor, e cada nó pode receber requisições *http* dos clientes.
- API Rest: Usamos o módulo *http.server* do python para criar um servidor básico que aceita requisições POST para resolver Sudoku e GET para obter estatísticas e informações da rede. A API possui os seguintes endpoints:
 - POST /solve: Recebe um puzzle em formato JSON, resolve o puzzle e retorna a solução.
 - GET /stats: Retorna estatísticas da rede de nós que participam da resolução distribuída.
 - GET /network: Retorna informações sobre os nós na rede.
- Os métodos HTTP têm uma função callback que processa a informação desejada e em seguida retorna.
- Solução Distribuída: Para resolver de forma distribuída um Sudoku com mais de 1 nó, cada nó irá resolver um “sub-puzzle” diferente a partir da primeira entrada vazia do Sudoku recebido, cujo valor será preenchido com os valores possíveis (de 1 a 9) e os nós irão testar cada um com valores diferentes. Ao encontrar o valor correto repete-se este processo com as outras entradas.

A redistribuição e armazenamento das tarefas que estão a ser resolvidas ajudam na resiliência e na recuperação de dados, caso por exemplo um nó saia da rede a tarefa em que o mesmo estava a trabalhar está armazenada e um nó que terminou a sua tarefa pode resolver a tarefa do outro que ainda não foi concluída.

A tarefa de um nó só é removida da lista de trabalhos em execução quando o mesmo entrega a sua solução, seja ela a solução do puzzle ou não.





Puzzle 1

-	-	-	-	-	-	-	-	-	-	-	-	-
1	6	3	8	9	2	7	5	4				
8	5	2	7	4	1	9	6	3				
7	4	9	5	3	6	2	8	1				
9	8	7	2	5	4	3	1	6				
5	2	6	1	7	3	4	9	8				
3	1	4	0	6	8	5	7	2				
6	0	0	4	2	5	8	3	7				
4	7	8	3	1	9	6	2	5				
2	3	5	0	0	7	1	4	9				

Puzzle 2

-	-	-	-	-	-	-	-	-	-	-	-	-
8	1	6	0	7	4	3	9	2				
3	5	2	9	8	1	6	7	4				
9	4	7	3	6	0	5	1	8				
6	0	5	1	4	8	2	3	7				
7	2	1	6	5	3	4	8	9				
4	8	3	7	2	9	1	6	5				
5	6	0	2	1	7	8	4	3				
1	7	8	4	3	5	0	2	6				
2	3	4	8	0	0	7	5	1				

Puzzle 1

Number of Nodes	Time to Solve	Total Validation
1 Node	134.47922587394714 s	590
2 Node	80.50234913825989 s	739
3 Node	65.38809204101562 s	920

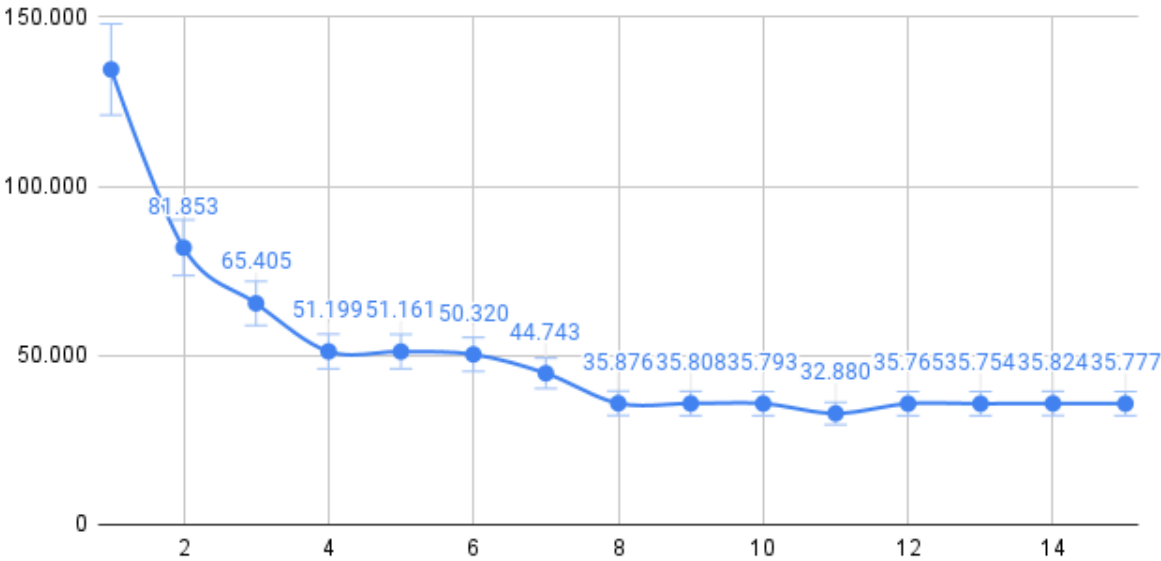
4 Node	50.270976543426514 s	1003
5 Node	50.26977181434631 s	1186

Puzzle 1

Number of Nodes	Time to Solve	Total Validation
1 Node	70.03671264648438 s	304
2 Node	53.49335551261902 s	477
3 Node	42.29222583770752 s	650
4 Node	39.723355531692505 s	823
5 Node	39.53257727622986 s	995

Referente ao puzzle 1

Relação Nós tempo



```
1 [{"all": {"solved": 12, "validations": 13297}, "nodes": [{"address": "192.168.41.54:7000", "validations": 2806},
2 {"address": "192.168.41.54:7001", "validations": 2540}, {"address": "192.168.41.54:7002", "validations": 1142},
3 {"address": "192.168.41.54:7003", "validations": 1354}, {"address": "192.168.41.54:7004", "validations": 2170},
4 {"address": "192.168.41.54:7006", "validations": 413}, {"address": "192.168.41.54:7007", "validations": 1290},
5 {"address": "192.168.41.54:7008", "validations": 590}, {"address": "192.168.41.54:7009", "validations": 794},
6 {"address": "192.168.41.54:7010", "validations": 198}, {"address": "192.168.41.54:7011", "validations": 0}]]
```

Interpretação dos Resultados

- **Região de Alta Eficiência (1-4 nós):** A diminuição rápida no tempo de resolução indica que o paralelismo inicial é altamente eficiente. A divisão do trabalho entre os nós reduz significativamente o tempo de computação.
- **Região de Ganhos Decrescentes (4-8 nós):** A taxa de diminuição do tempo de resolução começa a desacelerar. Isso sugere que o overhead de comunicação e coordenação entre os nós começa a compensar os benefícios do paralelismo adicional.
- **Região de Estabilização (8-15 nós):** O tempo de resolução se estabiliza, mostrando que o sistema atingiu um ponto de saturação. A adição de mais nós não melhora significativamente a eficiência, possivelmente devido ao aumento do overhead de coordenação e comunicação.

Conclusões

1. **Eficiência Inicial:** O aumento do número de nós de 1 para 4 resulta em uma melhoria significativa na eficiência, com uma redução drástica no tempo de resolução.
2. **Limites de Paralelismo:** A partir de 8 nós, os ganhos em eficiência se estabilizam. Isso indica que o sistema tem um limite de paralelismo além do qual os benefícios adicionais são mínimos.
3. **Overhead de Comunicação:** O overhead de comunicação e coordenação entre os nós começa a influenciar negativamente a eficiência quando o número de nós excede 4, tornando-se mais pronunciado após 8 nós.

Nossa análise final

1. **Número Ótimo de Nós:** Para maximizar a eficiência, o número de nós deve ser ajustado para cerca de 4 a 8. Este intervalo oferece um bom equilíbrio entre paralelismo e overhead de comunicação.
2. **Monitoramento de Overhead:** Monitorar o overhead de comunicação e coordenação é essencial para identificar o ponto de saturação e ajustar dinamicamente o número de nós conforme necessário.