

RELATÓRIO DO TRABALHO — PROGRAMAÇÃO CONCORRENTE E DISTRIBUÍDA

Nome: Jayme Pereira de Oliveira Neto

Matrícula: UC25102170

Disciplina: Programação Concorrente e Distribuída

Curso: Ciência da Computação

1- EXPLICAÇÃO TEÓRICA:

O que é Computação Distribuída?

A computação distribuída refere-se ao modelo de execução de software onde várias máquinas, fisicamente distribuídas, compartilham recursos, responsabilidades e tarefas. Em um sistema distribuído, o poder de computação não está concentrado em um só ponto; ele é compartilhado entre diferentes computadores que se comunicam para atender a um objetivo comum.

Isso proporciona maior eficácia, resiliência e velocidade de execução, sendo particularmente útil para problemas que exigem alto poder de processamento ou compartilhamento de recursos geograficamente distribuídos.

Referência: Tanenbaum, Andrew S.; Van Steen, Maarten. *Distributed Systems: Principles and Paradigms*. Pearson, 2007.

Escalabilidade e Tolerância a Falhas:

A escalabilidade refere-se à capacidade de um sistema distribuído de atender a um número maior de solicitações ou de dados, aumentando o número de servidores envolvidos.

Isso quer dizer que o modelo distribuído consegue adaptar o seu poder de execução ao crescimento da carga de trabalho, aumentando o hardware gradualmente.

A tolerância a falhas significa que o modelo consegue continuar funcionando mesmo que um ou mais servidores falhem, evitando a interrupção total do serviço.

Isso proporciona maior confiabilidade, sendo particularmente importante em aplicações que exigem alta disponibilidade.

Vantagens e Desvantagens:

1- Vantagens:

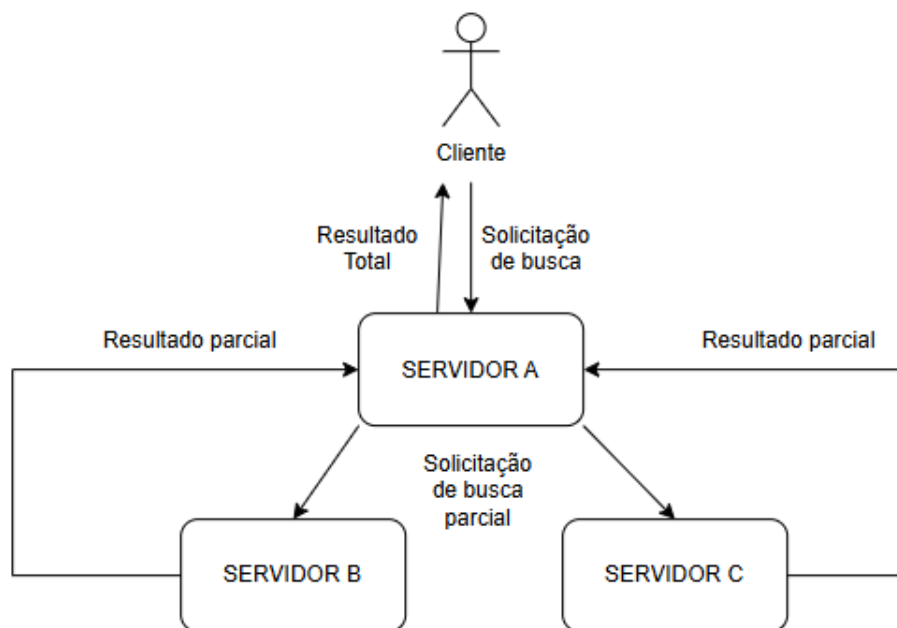
- Melhor compartilhamento de recursos.
- Alta escalabilidade.
- Tolerância a falhas.
- Melhor desempenho pelo paralelismo.

2- Desvantagens:

- Implementação mais complexa.
 - Aumento da latência na comunicação.
 - Dependência de redes e protocolos.
 - Sincronização de dados pode se tornar um problema.
-

2- ARQUITETURA DA SOLUÇÃO

Diagrama de comunicação:



Formato de dado trafegado:

- A comunicação é baseada em sockets TCP, utilizando BufferedReader e PrintWriter.
- Os dados trafegados serão strings UTF-8.
- A solicitação é uma única palavra-chave ou expressão, que o cliente quer encontrar no título ou na introdução do artigo.
- A resposta consiste em uma lista de artigos (cada um sendo um objeto JSON com "title" e "abstract"), concatenada como JSON para facilitar o parse pelo cliente.

Justificativa do algoritmo de busca:

Neste trabalho, foi usado o algoritmo de busca ingênuo (Naive).

Isso porque ele verifica a ocorrência da expressão de busca como uma substring em cada artigo.

Seu custo é $O(n * m)$, sendo n o tamanho do texto e m o tamanho do padrão.

Isso é suficiente para o conjunto de dados pequeno utilizado (algumas centenas de registros) e evita a complexidade de implementar algoritmos mais avançados como KMP ou Rabin-Karp.

Portanto, ele proporciona simplicidade na implementação e eficácia adequada para o tamanho de dados trabalhado.

3- Referências Bibliográficas

- TANENBAUM, Andrew S.; VAN STEEN, Maarten. *Distributed Systems: Principles and Paradigms*. Pearson, 2007.
- GEEKS FOR GEEKS. "Pattern Searching Algorithm." Disponível em: <https://www.geeksforgeeks.org/pattern-searching/>
- ORG.JSON. "JSON in Java." Disponível em: <https://www.devmedia.com.br/trabalhando-com-json-em-java-o-pacote-org-json/25480>