Católica
Universidade Católica de Brasília

Pró-Reitoria Acadêmica
Curso de Ciência da Computação
Trabalho de Programação
Concorrente e Distribuída

RELATÓRIO DO DESENVOLVIMENTO DO SISTEMA DE BUSCA DISTRIBUÍDO

Autores: Breno Gonçalves Gomes da Silva
Camila Mendes Osterno
Cariny Saldanha Oliveira
Ingrid de Oliveira Bonifacio

Orientador: João Robson Santos Martins

Brasília - DF
2025

1. INTRODUÇÃO

O seguinte relatório da disciplina de Programação Concorrente e Distribuída do curso de Ciência da Computação da Universidade Católica de Brasília (UCB), tem como objetivo descrever e esclarecer os aspectos do desenvolvimento de um sistema de busca distribuído. Este sistema contempla os assuntos abordados durante o período letivo da disciplina em uma aplicação prática contextualizada com o mundo real. Em especial será utilizado conceitos de computação distribuída, o uso de threads e sockets, tudo isso dentro da linguagem de programação Java.

Para isso foi utilizado o material previamente disponibilizado pelo professor, assim como, diversos meios de pesquisa que possam corroborar para o desenvolvimento do projeto tais quais como livros, vídeos e sites especializados, além do uso de ferramentas como o Eclipse, ChatGPT, entre outros.

2. FUNDAMENTAÇÃO TEÓRICA

Antes de prosseguir com o desenvolvimento do sistema é importante definir com clareza certos conceitos que, embora sejam aplicados e amplamente utilizados por usuários diariamente, não são senso comum para a maioria das pessoas. Com sorte os conceitos aqui abordados ajudem a compreensão do sistema desenvolvido.

3. COMPUTAÇÃO DISTRIBUÍDA

O escopo do projeto define que serão representados três servidores e entre eles será realizado um esforço computacional para resolver um problema para o usuário, que neste caso é a busca de um termo digitado por ele nos servidores. Em uma definição mais formal, Tanenbaum e Steen (2007, p. 1) definem “Um sistema distribuído é um conjunto de computadores independentes que se apresenta a seus usuários como um sistema único e coerente”.

Ou seja, a computação distribuída permite que diferentes máquinas colaborem para resolver problemas complexos de maneira mais rápida e escalável. No caso deste projeto, a simulação da busca entre três servidores ilustra bem esse conceito, mostrando como sistemas

distribuídos podem melhorar o desempenho, a confiabilidade e a disponibilidade dos serviços oferecidos aos usuários.

4. ESCALABILIDADE

Atualmente o sistema representa apenas três computadores para o sistema de busca, no entanto, se hipoteticamente o volume de dados crescer exponencialmente, pode ser difícil para hardware disponível ser capaz de realizar a busca e consequentemente, será notado pelo usuário sua lentidão. Logo, no nosso contexto, a escalabilidade é a capacidade do sistema de lidar com o crescimento de trabalho e seu potencial para ser ampliado para acomodar esse crescimento (Spasojevic, 2022).

Sendo assim, uma possibilidade para aplicação seria adicionar mais servidores para processar outras partes do dataset. Por exemplo, em vez de dois servidores (B e C) fazendo a busca, o dataset poderia ser dividido em quatro partes e ter quatro servidores buscando em paralelo. Isso melhora o tempo de resposta e a capacidade de lidar com múltiplas requisições.

5. TOLERÂNCIA E FALHAS

A tolerância a falhas é um atributo essencial em sistemas distribuídos, pois garante que o sistema continue a operar corretamente mesmo quando uma ou mais de suas partes apresentam falhas. Essa característica visa manter a disponibilidade e a confiabilidade do serviço, mesmo diante de interrupções parciais.

No contexto deste projeto, a tolerância a falhas se aplica principalmente à comunicação entre os servidores. Como os servidores B e C são responsáveis por partes distintas do dataset e operam de forma independente, a falha de um deles não deve comprometer completamente a funcionalidade do sistema. O servidor A, que atua como coordenador, deve ser capaz de identificar falhas na comunicação com B ou C e adotar estratégias de contingência, como retornar ao cliente apenas os resultados disponíveis ou emitir uma mensagem de erro informativa. Implementar esse tipo de robustez aumenta a confiabilidade geral da arquitetura.

6. O SISTEMA DESENVOLVIDO

O sistema desenvolvido é uma aplicação prática baseada na arquitetura cliente-servidor utilizando computação distribuída com comunicação via sockets TCP. O cliente envia uma palavra-chave para o servidor A (coordenador), que a repassa aos servidores B e C. Esses servidores realizam buscas locais em seus arquivos de texto e retornam ao servidor A os resultados encontrados, que são então consolidados e enviados ao cliente.

7. ESTRUTURA DO CÓDIGO

O sistema foi desenvolvido em Java e está dividido em quatro arquivos principais:

- Cliente.java: Responsável por enviar uma palavra ao servidor A e exibir os resultados recebidos.
- ServidorA.java: Recebe a palavra do cliente e a envia aos servidores B e C. Consolida os resultados.
- ServidorB.java e ServidorC.java: Cada um realiza buscas em seus próprios arquivos de dados e responde ao servidor A.

A comunicação é feita com `ServerSocket`, `Socket`, `BufferedReader` e `PrintWriter`. As operações são realizadas de forma concorrente por meio de threads.

8. TESTES REALIZADOS

Foram realizados testes com diferentes palavras-chave para garantir que os servidores B e C retornassem corretamente as ocorrências locais. Também foram simuladas falhas de rede (como a ausência de um dos servidores) para verificar o comportamento do servidor A diante da indisponibilidade parcial.

9. DIFICULDADES ENFRENTADAS

As principais dificuldades envolveram a sincronização entre os servidores e o tratamento de exceções durante a comunicação via sockets. Além disso, houve desafios na coordenação do recebimento e junção dos dados no servidor A, garantindo que a resposta ao cliente fosse clara e completa.

12. FORMATO DO DADO TRAFEGADO

O sistema utiliza **strings no formato de texto simples** como formato de dado para comunicação entre os componentes. O tráfego segue o seguinte padrão:

- **Cliente → Servidor A:**
  O cliente envia uma **palavra-chave** como string (ex: "banana"). Esta string representa o termo que será buscado nos arquivos.
- **Servidor A → Servidores B e C:**
  O servidor A replica a palavra-chave exatamente como foi recebida e a envia aos servidores B e C por meio de sockets.
- **Servidores B e C → Servidor A:**
  Cada servidor realiza a busca localmente em seus arquivos. Eles retornam uma **string contendo as linhas** onde a palavra foi encontrada ou uma mensagem como "Nenhuma ocorrência encontrada".
- **Servidor A → Cliente:**
  O servidor A concatena os resultados recebidos e envia ao cliente uma **única string formatada** com todas as ocorrências encontradas ou uma mensagem padrão de que nada foi localizado.

13. EXPLICAÇÃO E JUSTIFICATIVA DO FORMATO DE BUSCA

O algoritmo utilizado nos servidores B e C é a **busca sequencial (linear)** por correspondência de string.

Como funciona:

- O servidor abre seu próprio arquivo de texto.
- Ele percorre **linha por linha** o conteúdo do arquivo.

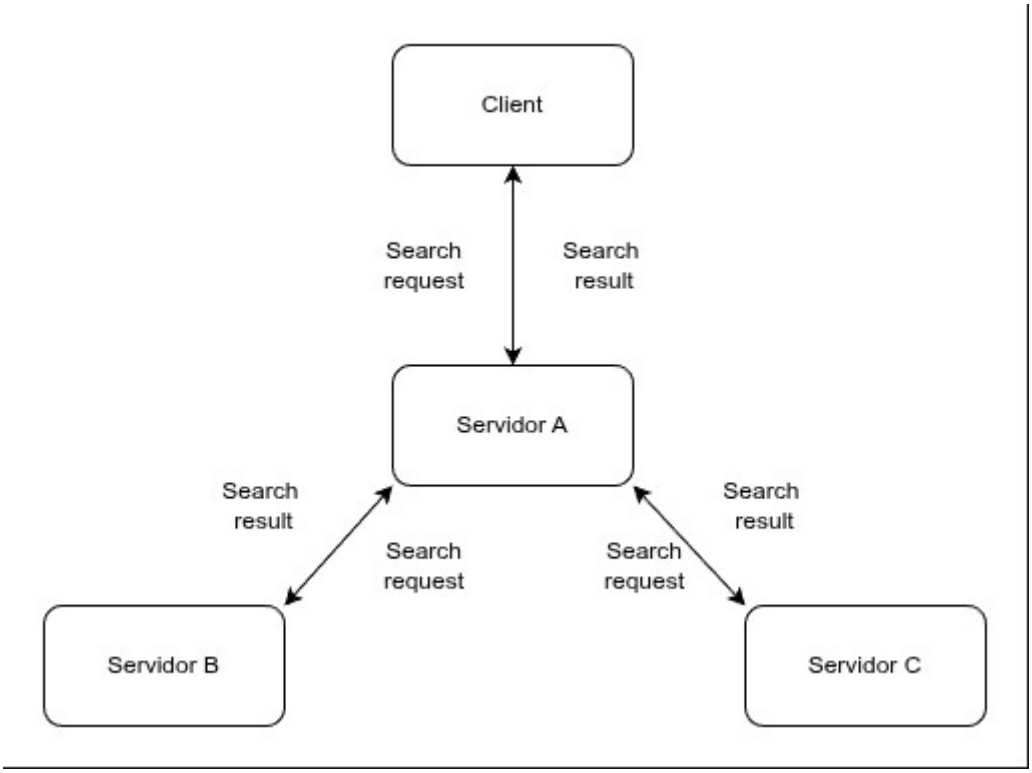
- Verifica se a **palavra-chave** está contida em cada linha.
- As linhas com correspondência são armazenadas como resultado.

Justificativas para a escolha:

- É simples de implementar e entender;
- Funciona bem com arquivos pequenos e médios;
- Adequado ao modelo distribuído — cada servidor pode buscar em paralelo;
- Não exige estrutura de dados complexa, nem ordenação.

Apesar de não ser o mais rápido para grandes volumes de dados, o algoritmo cumpre muito bem seu papel neste projeto, destacando os conceitos principais de concorrência, distribuição e comunicação entre processos.

14. DIAGRAMA



15. CONCLUSÃO

O projeto permitiu aplicar, de forma prática, os conceitos de computação distribuída abordados na disciplina. Por meio da implementação de uma arquitetura cliente-servidor com múltiplos servidores de busca, foi possível observar benefícios como paralelismo,

modularidade e divisão de carga. Também foram enfrentados desafios reais como sincronização, falhas de rede e tratamento de exceções. A experiência demonstrou como a computação distribuída pode ser aplicada para criar sistemas eficientes e escaláveis.

16. REFERÊNCIAS BIBLIOGRÁFICAS

TANENBAUM , Andrew S.; STEEN, Maarten Van. Sistemas Distribuídos: princípios e paradigmas. 2. ed. São Paulo: Pearson Universidades, 2007. 416 p. ISBN 8576051427.

O QUE é escalabilidade?. In: SPASOJEVIC, Anastasia. O que é escalabilidade?. [S. l.]: PhoenixNAP, 28 set. 2022. Disponível em: https://phoenixnap.pt/gloss%C3%A1rio/escalabilidade. Acesso em: 9 jun. 2025.