

Leitura de arquivo para contagem de palavras

Davi dos Santos Mattos
Daniel Li Vam Man
Pedro André Alves Chaves

Relatório Parcial
Programação Concorrente (ICP-361) - 2025/2

1. Descrição do problema geral

O problema escolhido é a contagem de palavras em um *arquivo.txt*. Uma tarefa muito comum em aplicações de análise de dados, mineração de texto, recuperação da informação e processamento de linguagem neural.

Uma palavra consiste numa sequência de caracteres do alfabeto e a contagem ignora se a letra está em maiúsculo ou minúsculo. As pontuações e espaços indicam o começo e fim de uma palavra.

- **Entrada:** Arquivo de Texto (.txt) qualquer.
- **Saída:** Número de palavras encontradas no arquivo

O problema se beneficia da concorrência quando temos que lidar com arquivos de texto muito grandes, pois podemos dividir o arquivo em partes e cada thread executará sua tarefa de contagem de forma paralela, reduzindo dessa forma o tempo de processamento.

2. Projeto e implementação da solução concorrente

Descrever o projeto da solução concorrente para o problema:

- apontar ajustes feitos no projeto inicial (se for o caso);
- descrever as principais decisões de implementação adotadas e suas justificativas.

Não mostrar o código-fonte, ele será visto diretamente. O objetivo aqui é descrever e justificar as decisões de projeto e implementação tomadas.

3. Testes de corretude

Descrever como o programa foi testado:

- descrever o conjunto de casos de teste usados para avaliação da corretude da solução proposta (lembrar de variar a dimensão dos dados de entrada e o número de threads usadas) e os resultados obtidos.

Não mostrar as telas de execução, o objetivo aqui é descrever como os testes foram feitos e os resultados obtidos.

4. Avaliação de desempenho

Descrever como o ganho de desempenho foi avaliado:

- descrever o conjunto de casos de teste usados para a avaliação de desempenho (lembrar de variar a dimensão dos dados de entrada e o número de threads usadas);
- descrever a configuração da máquina onde os testes foram realizados (identificação do processador, quantidade de núcleos de execução, sistema operacional);
- descrever quantas vezes o mesmo caso de teste foi executado e qual medida de tempo foi escolhida;
- mostrar o cálculo da aceleração e eficiência e/ou outras métricas pertinentes ao problema;
- apresentar os resultados obtidos condensados na forma de tabelas ou gráficos.

Não mostrar as telas de execução, o objetivo aqui é apresentar os resultados finais obtidos (tempo de execução, aceleração, eficiência, etc.).

5. Discussão

Apresentar uma análise dos resultados obtidos:

- discutir se o ganho de desempenho — ou as vantagens pretendidas com a solução concorrente — foram alcançadas ou não e por quais motivos;
- apresentar possíveis melhorias do programa, se for o caso;
- discutir outras questões que forem pertinentes;
- descrever dificuldades encontradas para a realização do trabalho, se for o caso.

6. Referências bibliográficas

References

- [1] Rossetto, S. **Slides de Aula**.
- [2] Maratona de programação paralela (Mackenzie). Disponível em:
<http://lspd.mackenzie.br/marathon/old.html>.
- [3] P. Pacheco, **An Introduction to Parallel Programming**, Morgan Kaufmann, 2011.
- [4] Martin Porter's Stemming algorithm as a C library. Disponível em:
<https://github.com/woorm/stmr.c>