Projeto da Disciplina

Versão 2024

INTRODUÇÃO

O problema de se buscar uma *string* em um conjunto de *strings* é amplamente estudado na literatura. Nele, dado uma *string* s[] e um texto T[], é necessário identificar quantas vezes a *string* ocorre no texto. Este problema possui alto custo computacional em determinadas situações, a depender do tamanho do texto em questão. No caso de grandes buscadores, a quantidade de texto pode ser superior a TB tendo assim uma necessidade de desenvolver algoritmos otimizados para este propósito. Um exemplo de aplicação é a busca de sequências genômicas em genomas.

O projeto da disciplina consiste na melhoria de desempenho de um algoritmo de localização de uma cadeia de bases nitrogenadas (cadeia de caracteres), armazenada no *arquivo1,* dentro das sequências armazenadas no *arquivo2, apresentando como resultado o total de ocorrências.* Cada sequência genômica é uma string delimitada por fim de linha (\n), exceto a última que não contém '\n'. É fornecido um algoritmo base sequencial que realiza a leitura linha a linha do arquivo2. Deverá ser utilizada a **linguagem C com o OpenMP** (não serão aceitas outras linguagens).

Exemplo:

Busca da cadeia "ACTTTGG"

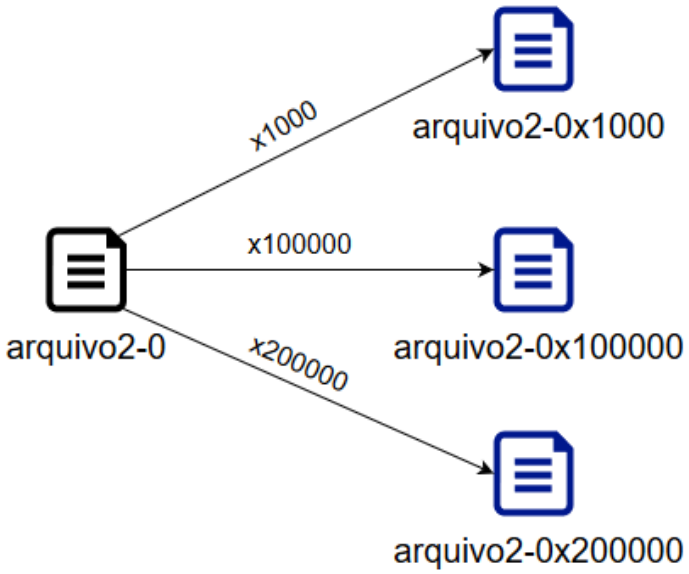
ATTGAACTTTGGTTAGCACTTTGGAATTC

ATCGTAGACTTTGGACTGA

Total de ocorrências: 3

O projeto consiste em:

1. Paralelização do programa sequencial fornecido (busca-cadeia.c).
2. Proposta e implementação de uma nova versão sequencial, visando melhorar o desempenho da execução. Uma estratégia é efetuar uma leitura única, em que são lidas todas as sequências do arquivo2, sendo estas mantidas na memória. Podem ser aplicadas também outras melhorias.
3. Paralelização da nova versão sequencial.
4. Montar tabelas ou gráficos com os tempos de execução das versões sequenciais e paralelas para:
   a. Variação de tamanho do arquivo2 e uso de 4 threads nas versões paralelas: usar arquivo2-0, arquivo2-0 vezes 1.000, arquivo2-0 vezes 100.000 e arquivo2-0 vezes 200.000. Para obter estes arquivos executar o arquivo fornecido denominado *aumentar-arquivo*, alterando-o para o número de vezes desejado. A figura a seguir ilustra esse processo.



   b. Variação do número de *threads* (1, 2, 4, 8), utilizando o arquivo2 com tamanho 200.000 vezes o arquivo2-0.

OBS: É importante apresentar o tempo de execução a partir de uma média de *n* tempos (por exemplo, 10) e o respectivo desvio padrão.

5. Análises de desempenho, considerando as versões sequenciais e paralelas.

Além da melhoria referente à leitura das sequências do arquivo 2 mencionada anteriormente outras poderão ser efetuadas. Como exemplo de melhorias no algoritmo fornecido pode-se buscar algoritmos sequenciais mais otimizados para este propósito e otimizações para aproveitar a hierarquia de memória do computador e otimizar o acesso aos dados do arquivo. A paralelização do algoritmo deverá respeitar a versão do OpenMP instalada na máquina remota, cujos comandos foram apresentados nas aulas teóricas.

Será realizada também para este projeto uma competição entre os grupos, onde uma parcela da nota do projeto será dada em função de um ranqueamento feito entre os grupos considerando o tempo de execução.

ENTREGA

Data de entrega: 03/12 (23:50)

Cada grupo deverá entregar o código da versão paralela do programa sequencial fornecido, o novo código sequencial e o seu respectivo código contendo a solução paralelizada. O relatório deverá descrever as estratégias adotadas na paralelização do programa sequencial fornecido e as aplicadas na nova versão sequencial e na sua respectiva paralelização.

A competição entre os grupos ocorrerá no dia 05/12, onde todos os códigos serão executados na máquina remota para a medição dos tempos de execução.

OBS: verificar a corretude do resultado comparando com o obtido executando a versão sequencial fornecida.

PROGRAMAS E ARQUIVOS DISPONÌVEIS

Encontram-se no Moodle os seguintes programas e arquivos:

- **busca-cadeia.c**: versão sequencial da aplicação
- **arquivo1**: contém a cadeia de bases nitrogenadas (cadeia de caracteres) a ser encontrada.
- **arquivo2-0**: arquivo com 64 sequências a ser utilizado para gerar o arquivo2.
- **aumentar-arquivo**: script para expandir um arquivo 1000 vezes. Para executá-lo:

  ```
  ./aumentar-arquivo arquivo2-0 arquivo2
  ```

  OBS: caso queira um número de vezes diferente, altere o código do arquivo *aumentar-arquivo* especificando o número de vezes desejado.