



Prof: Nilton Luiz Queiroz Junior  
Disciplina: Programação Concorrente (5205 – Turma 31)

## Primeiro Trabalho Prático

**Objetivo:** O trabalho tem como objetivo a implementação de uma aplicação paralela, em um ambiente de memória compartilhada centralizada e a elaboração de um artigo, fazendo análises de desempenho da aplicação implementada.

### Instruções:

- O trabalho poderá ser feito em dupla;
- A implementação da aplicação deverá ser feita na linguagem C/C++, Java ou Python, utilizando a biblioteca *pthread*s ou então a biblioteca padrão disponibilizada pela linguagem para implementação de *threads*.
  - Caso a dupla deseje utilizar outra linguagem/biblioteca deverá conversar com o professor;
- O trabalho tem valor de 0 a 10 e corresponde à primeira avaliação periódica da disciplina;
- Deverão ser entregues os seguintes itens:
  1. O código fonte do trabalho;
    - Tanto da versão sequencial quanto da paralela;
  2. Um passo a passo de como compilar e executar o programa;
    - Como deve ser fornecida a entrada para o programa, como escolher a quantidade de *threads* a serem usadas, qual sistema operacional a ser usado, e o que precisa ser instalado, como instalar e como configurar, etc;
    - **O passo a passo deverá explicar como compilar e executar por linha de comando, sem que seja necessário a instalação de uma IDE específica para tal fim.**
  3. O artigo, no modelo para a publicação de artigos da SBC, de 4 a 8 páginas, fazendo análise de desempenho composto por:
    - Título;
    - Resumo;
      - Não é obrigatório o *Abstract* (resumo em inglês);
    - Introdução;
    - Referencial teórico;
      - Descrever superficialmente técnicas que foram usadas na paralelização do algoritmo:
        - Por exemplo:
          - Descrever os conceitos de *threads* mecanismos de sincronização utilizados, etc;
    - Desenvolvimento;
      - Explicar como funciona a aplicação e como ela foi paralelizada;
      - Como foi feita o balanceamento da carga trabalho e justificar o por que da escolha;
      - Entre outros detalhes;



Prof: Nilton Luiz Queiroz Junior  
Disciplina: Programação Concorrente (5205 – Turma 31)

- Ambiente experimental e experimentos realizados;
  - Descrever hardware e sistema operacional utilizado;
  - Descrever detalhes dos experimentos que foram realizados;
    - Tamanho das entradas,
    - Quantidade de *threads*
    - Parâmetros do algoritmo;
    - Quantas execuções foram realizadas para cada variação do experimento;
- Análise e discussão dos resultados;
  - Extrair métricas tais como *speedup*, entre outras vistas em sala, e discutir os resultados com base nas métricas obtidas tais métricas;
    - Colocar gráficos (ou tabelas) com *speedup*, e as demais métricas, por quantidade de *threads*;
    - Tentar encontrar relações entre os resultados obtidos e a teoria vista em sala;
    - Em casos de resultados que não tem o comportamento esperado, tentar levantar hipóteses para justificar;
- Conclusões;
- Link para o download dos templates de modelo de artigo da SBC:
  - <http://www.sbc.org.br/documentos-da-sbc/category/169-templates-para-artigos-e-capitulos-de-livros>
- Os itens acima deverão ser entregues via moodle, em um arquivo compactado que deve seguir o padrão:
  - NomeAluno1\_RAXXXXXX\_NomeAluno2\_RAXXXXXX;
    - O formato do arquivo para submissão poderá ser .ZIP, .RAR ou .TAR
- A data limite para a entrega do trabalho será combinada em sala de aula;

**Descrição:** Neste trabalho deverá ser implementado o algoritmo K-means, fazendo a paralelização dos cálculos realizados sobre os exemplos da base de dados. Serão disponibilizadas 5 bases. Cada base tem uma quantidade diferente de atributos, porém todas elas seguem o mesmo padrão:

- Cada linha representa um exemplo;
- Todos os atributos de cada exemplo são números inteiros;
- Os atributos são todos separados por vírgula;

Além das bases, também serão disponibilizados os arquivos contendo os atributos dos centroides iniciais, que seguem o mesmo padrão.

As bases tem a seguinte nomenclatura:

int\_base\_<numero\_de\_atributos>.data

Os arquivos com os centroides tem a seguinte nomenclatura:

int\_centroides\_<numero\_de\_atributos>\_<numero\_de\_centroides>.data

Por exemplo:

int\_base\_256.data

Esse arquivo indica que a base tem 256 atributos, ou seja, são 256 números inteiros os quais deverão ser usados para a clusterização.

int\_centroides\_256\_20.data



Prof: Nilton Luiz Queiroz Junior  
Disciplina: Programação Concorrente (5205 – Turma 31)

Esse arquivo indica que existem 20 centroides com 256 atributos (consequentemente, estes centroides devem ser utilizados junto a base de 256 atributos).

Para um melhor entendimento do algoritmo K-means, a seguir serão apresentados alguns conceitos de aprendizagem de máquina e uma pequena descrição do algoritmo.

Um dos principais objetivos da maioria dos algoritmos de aprendizagem de máquina é classificar exemplos baseados em seus atributos. Dentro da área de aprendizagem de máquina existem algoritmos que tentam agrupar os exemplos levando em consideração treinamento, os quais dependem de exemplos previamente classificados e são conhecidos como algoritmos de classificação supervisionada. Além destes, existem outros algoritmos que tentam agrupar os exemplos, sem a necessidade de exemplos prévios, os quais são chamados de algoritmos de classificação não supervisionada.

Ambas os tipos de algoritmos levam em consideração um conjunto de características extraídos dos exemplos, tais características são chamadas de atributos.

Métodos de aprendizagem supervisionados necessitam de um atributo denominado classe, o qual determina o que se irá aprender. Estes métodos usam a classe de exemplos anteriores para prever a classe de um novo exemplo. Já métodos não supervisionados tentam agrupar os exemplos que tem maior nível de similaridade, pois grupos similares podem ser vistos como potenciais classes.

O algoritmo K-means é um algoritmo de classificação não supervisionada, e a ideia por trás de tal algoritmo é processar um conjunto de exemplos, de maneira a agrupá-los em K grupos distintos.

Porém, o algoritmo K-means não é capaz de decidir a quantidade de agrupamentos a serem gerados, então, a quantidade de agrupamentos é um parâmetro do algoritmo.

A ideia geral por trás do algoritmo K-means é utilizar K centroides, os quais serão as referências para cada um dos agrupamentos (os centroides **podem não** ser exemplos que estão nos agrupamentos).

Desta forma, o algoritmo irá executar e agrupar os elementos, associando cada exemplo ao centroide mais próximos, dando a todos eles o mesmo rótulo. Após rotular os exemplos é necessário reposicionar cada um dos centroides. O algoritmo irá então se repetir até que nenhum centroide sofra qualquer alteração. Um pseudocódigo para o algoritmo K-means é mostrado na figura 1. Sendo B a base de dados e C o conjunto de centroides (contendo k centroides).

```
K-means(B, C)
  Repita
    Para cada exemplo  $E \in B$ 
       $E.\text{rotulo} = \text{Rótulo do centroide } C_i \in C \text{ mais próximo de } E$ 
    Para cada Centroide  $C_i \in C$ 
      Recalcule as coordenadas de  $C_i$ 
  Até que não exista alteração em C
```

*Figura 1: Pseudo-código para o algoritmo K-Means*

O cálculo de distância e o procedimento de recalcular a posição dos centroides podem ser feitos de diversas maneiras. A mais comum para o cálculo de distâncias é a distância euclidiana, que é dada por:



Prof: Nilton Luiz Queiroz Junior  
Disciplina: Programação Concorrente (5205 – Turma 31)

Distancia Euclidiana entre dois pontos:

$$D(P, Q) = \sqrt{\sum_{i=1}^n (P_i - Q_i)^2}$$

Já o reposicionamento do centroide, é mais comumente calculado por meio das médias dos exemplos que possuem o mesmo rótulo do centroide. Dessa forma, para todos os pontos que possuem o mesmo rótulo do centroide, obtém-se a média de cada um dos atributos, e cada uma destas médias é o novo “atributo” do centroide.

Por exemplo:

Suponha uma base onde cada exemplo possui 4 atributos. Suponha também que temos um dos centroides  $C_x$  com as seguintes coordenadas: (20,37,43,58). Além disso, apenas três pontos tem  $C_x$  como o ponto mais próximo, e são eles:

$E_1 = (20,38,43,60)$ ;  $E_2 = (18,33,45,62)$ ; e  $E_3 = (25,37,44,64)$

Com isso, o novo centroide será calculado da seguinte maneira:

O primeiro “atributo” do centroide será realizado pela média aritmética do primeiro atributo dos pontos  $E_1$ ,  $E_2$  e  $E_3$ :  $(20 + 18 + 25)/3 = 21$

O segundo “atributo” do centroide será:  $(38 + 33 + 37)/3 = 36$

O terceiro “atributo” do centroide será dado por:  $(43 + 45 + 44)/3 = 42$

E o quarto “atributo” será dado por:  $(60 + 62 + 64)/3 = 62$

Portanto, o centroide  $C_x$  passará então a ser: (21, 36, 42, 62).

**Obs:** Devido à sensibilidade do algoritmo, quando um exemplo possuir a mesma distância para dois ou mais centroides, tanto na versão paralela quanto na sequencial, deve ser determinado o mesmo critério de desempate, para que não exista o risco de quantidades diferentes de iterações nas versões sequencial e paralela.

Além disso, por problemas de arredondamento de ponto flutuante é recomendado que ao calcular um centroide com base nos pontos mais próximos, o resultado gerado seja arredondado somente para cima ou somente para baixo, para evitar possíveis erros com cálculos de ponto flutuante.

## Problemas com Trabalhos COPIADOS:

Quem copiar terá o trabalho anulado (zerado), seja de outra dupla ou da internet.

Quem fornecer a cópia também terá o trabalho anulado (zerado).

### Referências para o algoritmo de agrupamento:

Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. 2005. *Introduction to Data Mining, (First Edition)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Capítulo 8.2.

Ian H. Witten and Eibe Frank. 2005. *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition* (Morgan Kaufmann Series in Data Management Systems). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Capítulo 4.8.