



## INF1406 – 2016.1

### Exercício 2: Multiplicação Paralela de Matrizes

1. Nesse exercício, o objetivo é implementar um programa capaz de multiplicar várias matrizes de tamanho fixo,  $N \times N$ , realizando os cálculos independentes de forma paralela. O cálculo de matrizes é dado pela seguinte fórmula:

$$(AB)_{ij} = \sum_{k=1}^m A_{ik} B_{kj}.$$

2. O programa deve receber como parâmetros, nesta ordem:
  - a. O caminho para um arquivo com as matrizes
  - b. O tamanho das dimensões das matrizes ( $N$ )
  - c. O número máximo de *threads* a utilizar no cálculo ( $M$ )
  - d. O número de matrizes que o arquivo contém
3. Após a leitura dos parâmetros, o programa deve utilizar as  $M$  threads para realizar cada multiplicação de duas matrizes. Por “cálculos independentes” a serem realizados de forma paralela, você pode considerar a fórmula acima, que explicita o cálculo de um item da matriz. Portanto, cada cálculo desse tipo deve ser realizado em uma *thread* do conjunto de *threads* disponíveis.
4. O tipo dos números contidos nas matrizes deve ser *double*.
5. O arquivo com as matrizes deve ser texto puro e conter os valores das colunas separados por espaços brancos. Por exemplo, o conteúdo de um arquivo com três matrizes  $2 \times 2$  seria algo como:  
  
1.0887 59  
0.004 0.235890  
9858.01 1.3  
1 1000  
0 9  
98 5.5
6. Seu programa deve calcular o tempo total gasto no cálculo das matrizes e imprimir em tela.
7. Por fim, antes do término, deve escrever um novo arquivo de nome “resultado.txt” com o resultado final obtido, seguindo o mesmo formato do arquivo de entrada. Ou seja, se foram multiplicadas matrizes  $N \times N$ , o



arquivo “resultado.txt” deve conter uma única matriz  $N \times N$  com o resultado das multiplicações.

8. Seu trabalho deve contemplar:

- Um projeto Eclipse Java (ou um script responsável por compilar corretamente o código e outro responsável por executar o código).
- O código-fonte da sua solução. A linguagem utilizada deve ser Java, mas você pode utilizar outra de sua preferência, desde que se responsabilize por fazer o seu programa rodar na minha máquina, que conterà a JVM 1.6.

Observações:

- A JVM a ser utilizada deve ser a 1.6.
- O trabalho deve ser feito individualmente ou em duplas.
- O prazo do trabalho será definido em aula.
- Você pode enviar o seu trabalho por email para [caugusto@inf.puc-rio.br](mailto:caugusto@inf.puc-rio.br) ou [caeaugusto@gmail.com](mailto:caeaugusto@gmail.com).
- A apresentação será feita durante o horário de aula, na sala de aula. Caso não possa comparecer, você deve enviar o trabalho por email até as 19h do dia de entrega e agendar outro horário de apresentação (com perda de pontos).
- Cada dia de atraso incorre na perda de um ponto na nota. Cada dia de atraso na apresentação (entregando o trabalho até a data limite mas apenas não apresentando) incorre na perda de meio ponto na nota.
- Durante a apresentação, serão feitas perguntas para avaliar o seu entendimento do problema, da solução, dos erros de concorrência e suas soluções. A nota será baseada nas suas respostas e no fato dos seus programas funcionarem como especificado ou não.
- Não é objetivo do trabalho o estudo de algoritmos de multiplicação de matrizes. Portanto, qualquer solução para o problema será aceita, desde que realize os cálculos de forma paralela.
- Avalie a diferença no tempo total gasto de acordo com a quantidade de *threads*. Caso obtenha sempre resultados muito semelhantes, independente do número de *threads*, você precisará utilizar matrizes com  $N$  maior ou artificialmente gastar mais tempo em cada cálculo. Não utilize *Thread.sleep()* ou outra forma de bloqueio da *thread*, como *wait()* (por quê?). Você pode incluir um *loop* finito, por exemplo.