



Pontifícia Universidade Católica de São Paulo
Faculdade de Estudos Interdisciplinares - FACEI
Ciência de Dados e IA

Experimento 01

Sistemas Distribuídos e Computação Paralela

Prof. Carlos Paes.

Objetivo: Este experimento tem como o objetivo proporcionar ao aluno a prática da programação paralela por meio das etapas de modelagem do problema usando paralelismo SIMD e a implementação da solução na linguagem Python.

Experimento 01 – Multiplicação de Matrizes

A multiplicação de matrizes é uma operação muito utilizada na área de computação. Em computação gráfica e jogos os objetos gráficos no vídeo do computador são manipulados através da multiplicação de matrizes que representam transformações geométricas como: reflexões, contrações, rotações, projeções, translações etc. Enquanto muitas destas transformações são lineares, como por exemplo as reflexões, contrações e projeções, as translações e rotações fora da origem não são lineares ¹. Além da computação gráfica, multiplicação de matrizes são utilizadas na área de Inteligência Artificial. A maioria das operações de *deep learning* e *machine learning* são em vetores e matrizes. Nos links abaixo você encontrará as informações de como multiplicar matrizes. Fique à vontade para consultar outros sites sobre o assunto durante o laboratório.

- <https://www.todamateria.com.br/multiplicacao-de-matrizes/>
- <https://brasilecola.uol.com.br/matematica/multiplicacao-matrizes.htm>

Considerando a importância dessa operação matemática para a área de computação, cumpra as seguintes etapas propostas para o experimento.

Etapas I: Implementar um programa sequencial na linguagem Python para multiplicação de duas matrizes $N \times N$. Após a implementação faça os testes necessários.

¹ <https://youracademic.org/docs/4h>

Etapa II: Após a implementação faça uma análise do desempenho do algoritmo implementado e calcule o tempo para a computação da operação considerando uma matriz de dimensão 500X500. Inicialize com dados aleatórios antes do processamento.

Etapa III: Implementar um programa paralelo em Python usando Numba para multiplicação de duas matrizes $N \times N$. Utilize um conjunto de 4 threads para o processamento paralelo.

Etapa IV: Calcule o tempo para a computação da operação considerando uma matriz de dimensão 500X500 e conjuntos de 2, 4 e 9 threads. Inicialize com dados aleatórios antes do processamento.

Etapa V: Calcule o SpeedUp da solução paralela proposta na etapa III. Indique no final do cálculo as informações de configuração do hardware utilizado (CPU, Clock, Cache, memória RAM e SO).

Experimento 02 – Produto Escalar

Assim como a operação de multiplicação matricial, o produto escalar é operação matemática da álgebra linear amplamente utilizada na área de computação gráfica e jogos (ex: *shaders* são programas que utilizam o produto escalar para criar um efeito holográfico. Normalmente são executados pelas placas de vídeo ou GPUs. Nos links abaixo você encontrará as informações de como realizar o produto escalar. Fique à vontade para consultar outros sites sobre o assunto durante o laboratório.

- <https://www.matematica.pt/geogebra/11-ano-produto-escalar-vetores.php>
- <https://pt.khanacademy.org/math/multivariable-calculus/thinking-about-multivariable-function/x786f2022:vectors-and-matrices/a/dot-products-mvc>

Considerando a importância dessa operação matemática para a área de computação, cumpra as seguintes etapas propostas para o experimento:

Etapa I: Implementar um programa sequencial na linguagem Python que calcule o produto escalar entre dois vetores (u e v) de tamanho 100. Após a implementação faça os testes necessários.

Etapa II: Após a implementação faça uma análise do desempenho do algoritmo implementado e calcule o tempo para a computação da operação considerando dois vetores de tamanho 100. Inicialize com dados aleatórios antes do processamento.

Etapa III: Implementar um programa paralelo em Python usando Numba para calcule produto escalar de dois vetores de tamanho 100. Utilize um conjunto de 4 threads para o processamento paralelo.

Etapa IV: Calcule o tempo para a computação do produto escalar considerando 2, 4 e 8 threads. Inicialize com dados aleatórios antes do processamento.

Etapa V: Calcule o SpeedUp da solução paralela proposta na etapa III. Indique no final do cálculo as informações de configuração do hardware utilizado (CPU, Clock, Cache, memória RAM e SO).

Experimento 03 – Análise do Projeto Integrador

Neste semestre você está desenvolvendo o projeto integrador que envolve a implementação de algoritmo de aprendizado de máquina para predição de dados consumidos (via API) de um serviço gerenciado na Internet. A partir da solução proposta no projeto, faça as seguintes atividades:

- 1) Identifique no projeto a(s) parte(s) da implementação **que poderia(m) ser otimizada(s) usando Numba e o paradigma de computação paralela***. As seguintes etapas deverão ser cumpridas: (i) descrever as partes do projeto desenvolvido que poderiam ser otimizadas usando computação paralela. A descrição deverá ser compilada em um documento; (ii) implementar as partes do projeto identificados usando os recursos disponibilizados pelo módulo Python Numba; (iii)

realizar a análise de desempenho da proposta de otimização do código desenvolvido usando computação paralela e otimização de código; e (iv) complementar o documento elaborado na etapa (i) com os resultados da análise de desempenho realizada na etapa anterior.

- 2) Analisar se o algoritmo de aprendizado de máquina utilizado no projeto faz uso de algum recurso de computação paralela (paralelismo de dados ou tarefa). Deve-se tentar identificar qual o nível de paralelismo (instrução ou aplicação) e quais partes do algoritmo foram otimizadas por código nativo como, por exemplo, a solução jit Numba² ou CPython³ (chamada de código em C no Python).

* **IMPORTANTE:** No caso do item 1), pode-se chegar à conclusão que não é possível usar computação paralela para acelerar (SpeedUp) o código desenvolvido no projeto integrador. Neste caso o grupo deverá justificar de forma adequada.

Questões de Ordem

- Grupo de 3/4 alunos
- Implementar em Python e usando a biblioteca Numba
- Os experimentos deverão ser entregues no final da aula do **dia 12/04**
- Os arquivos com o código fonte e os cálculos deverão ser enviados na tarefa do Team. (IMPORTANTE: Não compactar os arquivos!)
- O grupo que copiar código da internet receberá nota 0. (será usado um programa para detecção de cópia)

² <https://numba.pydata.org/>

³ <https://cython.org/>