Pontificia Universidad
JAVERIANA
Cali

Facultad de Ingeniería y Ciencias
Departamento de Electrónica y Ciencias de la Computación

Carrera de Ingeniería de Sistemas y Computación

Programación Paralela

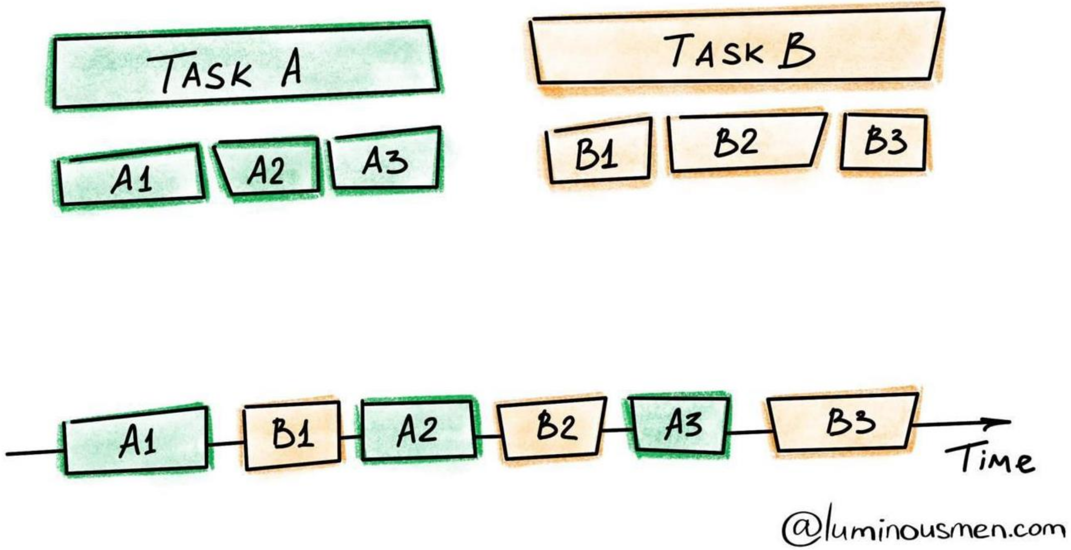
Leveraging Python scripts

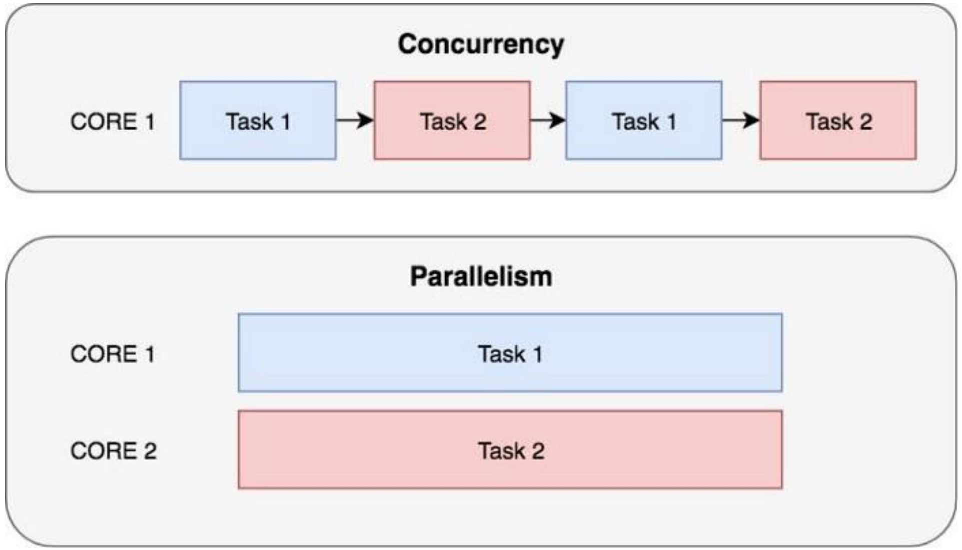
Mayo 2022

Tek

Python

- One of the top languages of choice for tasks relating to Machine Learning or Data Science in general:
  - Very high-level language
  - Interpreted language
  - Trade-off: development time vs slow execution speed
- How to leverage better execution speeds?





Leveraging Python

- Mind your Python code
- Concurrency
  - Using the `threading` module (*Global Interpreter Lock*, *GIL*, makes sure that at any given time, only one thread runs to handle thread-safe executions and to prevent race conditions… good for I/O-bound tasks)
  - Using the `multiprocessing` package. It supports spawning processes by using subprocesses instead of threads, allowing full leverage of multiple processors on a given machine
- Using optimised libraries
  - SciKits
    - NumPy: Powerful N-dimensional arrays, Numerical computing tools
    - SciPy library: numerical routines, such as routines for numerical integration, interpolation, optimization, linear algebra, and statistics
    - Matplotlib: a comprehensive library for creating static, animated, and interactive visualizations in Python
    - Ipython: provides a rich architecture for interactive computing
    - SymPy: a Python library for symbolic mathematics
    - pandas: a fast, powerful, flexible and easy to use open source data analysis and manipulation tool

Leveraging Python

- C extensions for Python (Cython, an optimising static compiler for both the Python programming language and the extended Cython programming language (based on Pyrex). It makes writing C extensions for Python as easy as Python itself
- Scaling out with mpi4py (MPI for Python)
- Using Numba, a Python compiler from Anaconda that can compile Python code for execution on CUDA-capable GPUs or multicore CPUs.
- Accelerating with PyCUDA*

*From CUDA 10.2 onwards, CuPy is also available. CuPy is an open-source array library for GPU-accelerated computing with Python.

Pre-requisites

- Install Anaconda Individual Edition (Python 3.8)
- `pip install mpi4py`
- `pip install pycuda`