1. For each of the following examples, identify the type of parallelism being used and explain your reasoning:
   * A Python script using the multiprocessing module to run 4 independent tasks.

This would be task-level parallelism. Since each task is independent, nether the instruction, nor the data can be assumed to be the same. As a result, we need a type of parallelism that uses MIMD. Looking at the different types of parallelism, Task-level parallelism meets this requirement.

* + Matrix multiplication accelerated with NumPy’s vectorized operations.

This type of parallelism is Data-level parallelism. Data-level parallelism is all about single instruction multiple data. Matrix multiplication is a task where there is a bunch of data that all needs to have the same operation performed on it.

* + A Raspberry Pi cluster using MPI to calculate parts of a large dataset.

This is Parallelism in Hardware. This uses the Multiprocessor type of hardware parallelism. Specifically, Distributed-memory multiprocessors. Each processor can only directly access memory in its own node and Communication is done through explicit message passing. MPI is the Message Passing Interface. According to Wikipedia, “[it] is a portable message-passing standard designed to function on parallel computing architectures. The MPI standard defines the syntax and semantics of library routines that are useful to a wide range of users writing portable message-passing programs.” In other words, MPI is for message passing. Message passing is central to distributed computing. Logically, this is distributed.

1. If your calculation rand sequentially in 18.0 ms and in parallel on 4 cores in 5.2 seconds
   * What is the speedup factor?
   * Is this an ideal speedup factor
   * What factors could prevent a perfect 4x improvement?