Exercises assignment #2 – parallel computing in Python

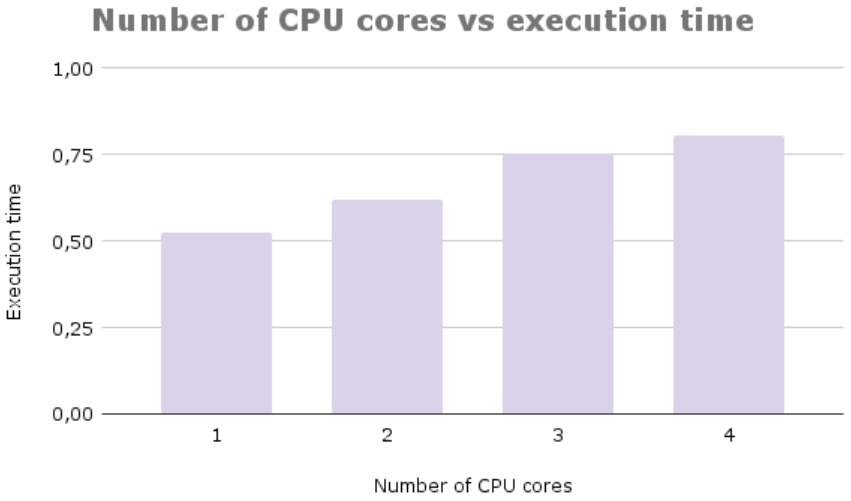
Disclaimer on decorators: I tried to time the count_properties function using a decorator, however the mapping functions requires that the mapped function is pickable (AttributeError: Can't pickle local object 'time_it.<locals>.wrapper_function'), which means that it can be serialized. Thus, I chose a simplier approach and timed it directly in the loop iterating over number of CPU cores.

The exemplary database in CSV format is Statistics on Maori businesses and was downloaded from: https://www.stats.govt.nz/large-datasets/csv-files-for-download/.

Use of AI: I used ChatGPT3.5 to find out more about multiprocessing framework and multi-core computations in Python. However, the code itself was written by me.

Results: Function for counting properties was performed 10 times for each number of CPU cores (from 1 to number of cores, which is returned by multiprocessing.cpu_count() function). Here are the obtained results:

```
Mean execution time for 1 core: 0.5249699010000768 s
Mean execution time for 2 cores: 0.6175291380004637 s
Mean execution time for 3 cores: 0.7537622340005328 s
Mean execution time for 4 cores: 0.8049154660002387 s
```



As we can observe, results do not vary a lot; execution time slightly increases with the growing number of cores used. The most likely reason for that is the low complexity of performed computations – joining concurrent threads takes time itself, thus it is not worth to explicitely use more than one core when not needed. Probably the database is too small for the difference to be seen.