Lab 1
Multithreading
Multiprocessing
Numba

Master 2 SID
Benoist GASTON
benoist.gaston@univ-rouen.fr

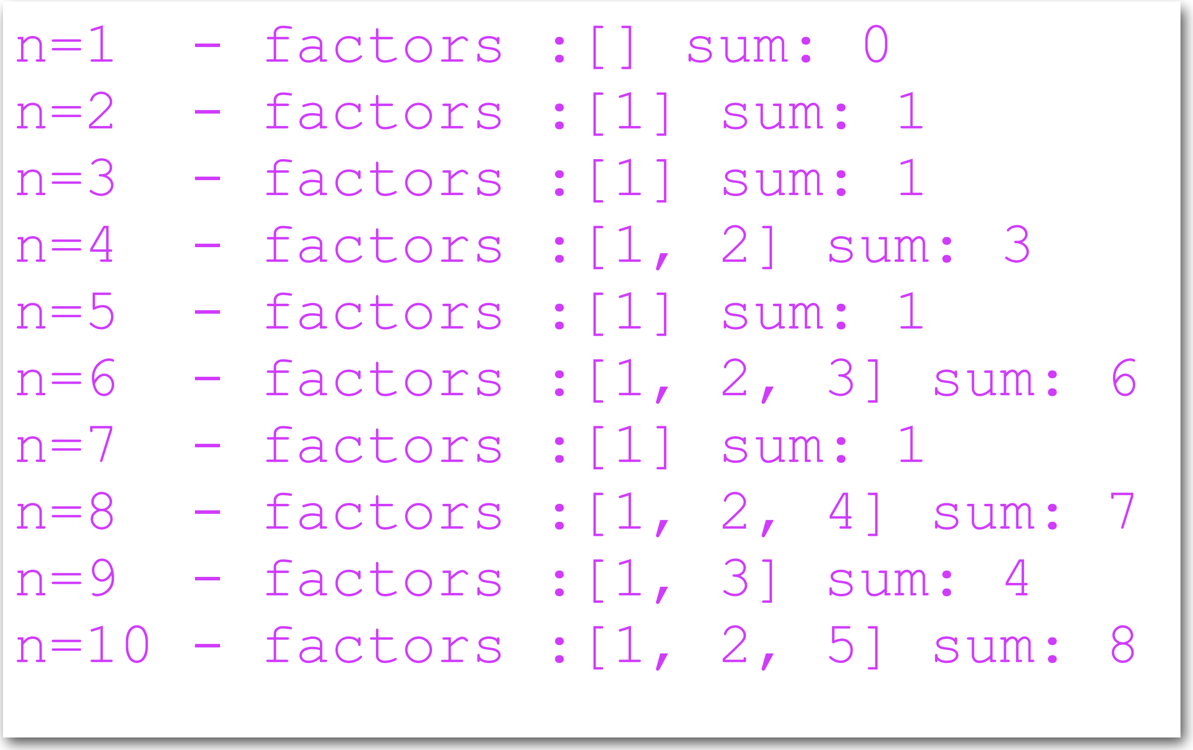
Introduction

- The code is available on github: https://github.com/gastoben/S3UE2_HPC/tree/main/TPs/TP01

- The function `factor_01(n)` of `TP01_01_factor.py` contient is a naive algorithm to construct by comprehension the list of all the integer factor of des différents facteurs entiers de `n` strictly less than `n`.

- The function `main(a, b)` in `TP01_01_factor.py` c construct by comprehension the list of sums of all the integer factors of each integer `n` between `a` and `b` given as arguments of the script.

- The lab consist of improving the performance using Numba and to paralyze it with multiprocessing and multithreading using `concurrent.futures`.



```
n=1  - factors :[] sum: 0
n=2  - factors :[1] sum: 1
n=3  - factors :[1] sum: 1
n=4  - factors :[1, 2] sum: 3
n=5  - factors :[1] sum: 1
n=6  - factors :[1, 2, 3] sum: 6
n=7  - factors :[1] sum: 1
n=8  - factors :[1, 2, 4] sum: 7
n=9  - factors :[1, 3] sum: 4
n=10 - factors :[1, 2, 5] sum: 8
```

factors from 1 to 10

Performance and Numba

preliminary questions

- Getting started with the code and et estimate performance on a single node.
- **Questions**
  1. Run the code with different value for `a` and `b`.
  2. Profile the code using `cProfile`, `%time`

Numba

- Test the just-in-time compilation with Numba. For the moment we will not release the GIL.
- **Questions**
  1. Add the `@jit` decorator the the function `factor01`. How does this impact performance?
  2. Add the `@jit` decorator the the function `main`. How does this impact performance?
  3. Add the option `nopython = True` to the `@jit` decorator (or use `@njit`). How does this impact performance?

Multitasks

Multiprocessing

For this part, we will use the class `ProcessPoolExecutor` of `concurrent.futures`

Questions

1. Remove the `@jit` decorator from the `main` function.
2. Modify the `main` function to use the Python's method `map`.
3. How does this impact performance?
4. Adapt the code in order to dispatch the tasks performed by `map` between the differents processes of a pool of `n` processes, `n` given as an argument of `main`.
5. How does this impact performance?

Multithreading

Pour cette partie, nous utiliserons la classe `ThreadPoolExecutor` de `concurrent.futures`

Questions

1. Adapt the multiprocess code in order to use `n` threads rather than n processes.
2. How does this impact performance?
3. Use the option `nogil = True` of the `@jit` decorator.
4. How does this impact performance?