CNCA
Colaboratorio Nacional de Computación Avanzada

KABRÉ SUPERCOMPUTER

python paralelo básico, un repaso general

Leonardo van der Laat

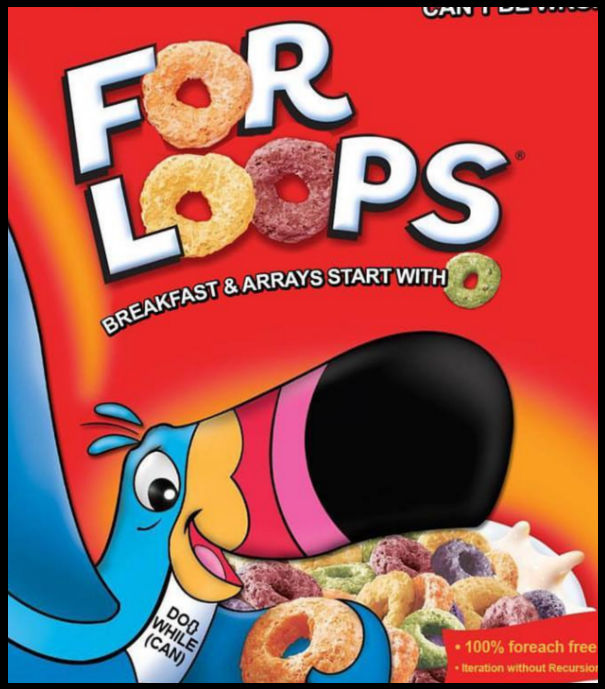
Sumario del repaso

- Optimizar antes de paralelizar
- Paralelismo *nativo* (biblioteca estándar)
  - `threading`
  - `multiprocessing`
- `pymp`, inspirado en OpenMP
- `mpi4py`, *patois* de python con `MPI`

Optimizar antes de paralelizar

y luego paralelizar para optimizar

Optimizar antes de paralelizar



vs

secuencial

Vectorial, BLAS

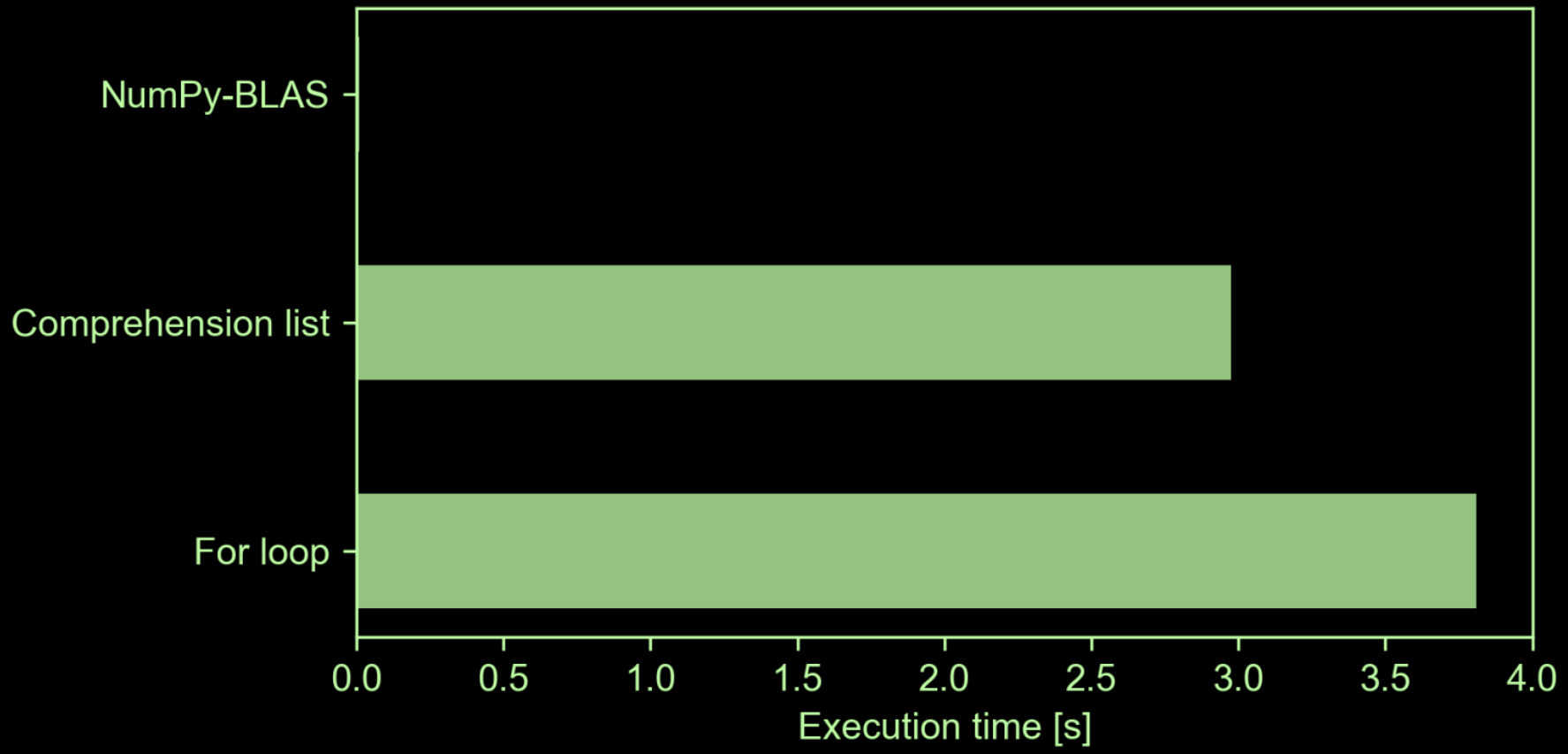
Producto escalar

```
# Fruit loops
dot_product = 0
for i in range(len(a)):
    dot_product += a[i] * b[i]


# Comprehension list
dot_product = sum(a_i*b_i for a_i, b_i in zip(a, b))


# NumPy (BLAS Linear Algebra Subprograms)
dot_product = np.dot(a, b)
```

Tiempo de ejecución




gettyimages
25
Dorling Kindersley

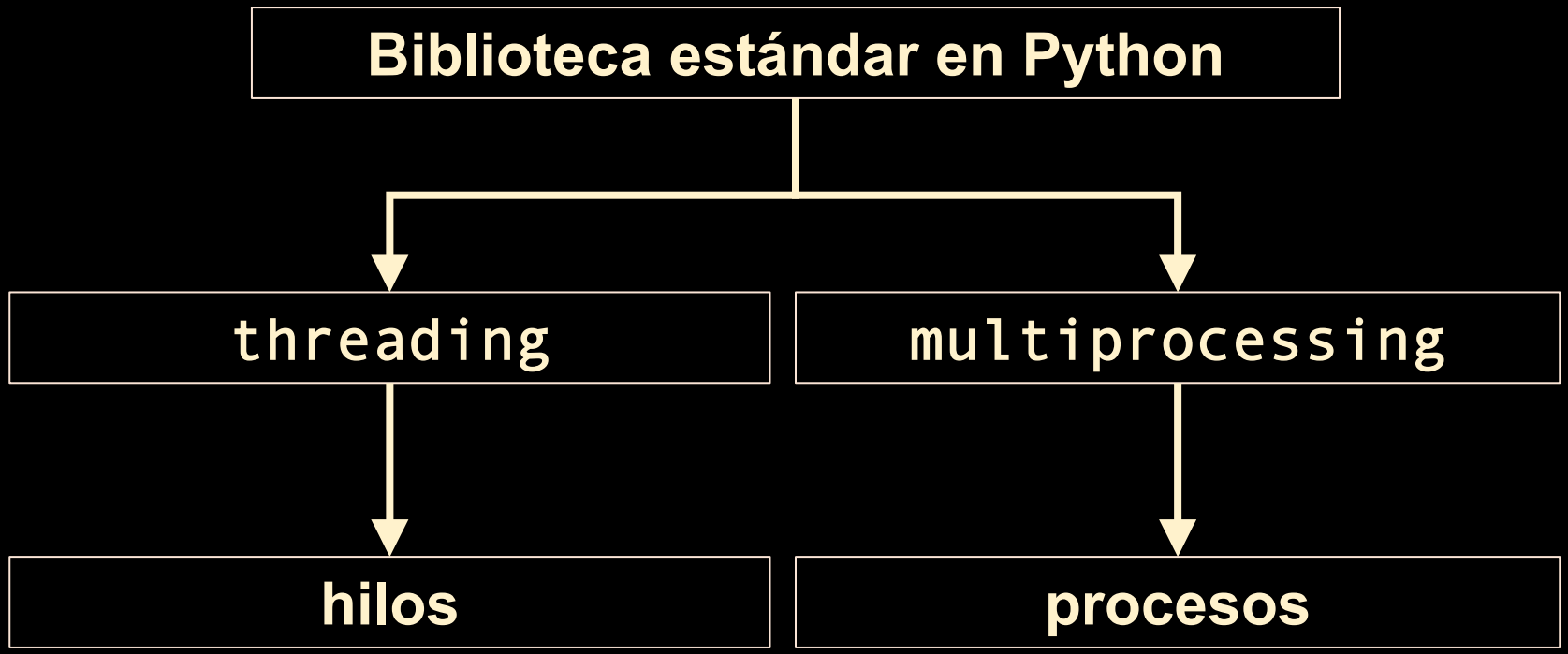
Paralelismo ”endémico”

Biblioteca Estándar de Python

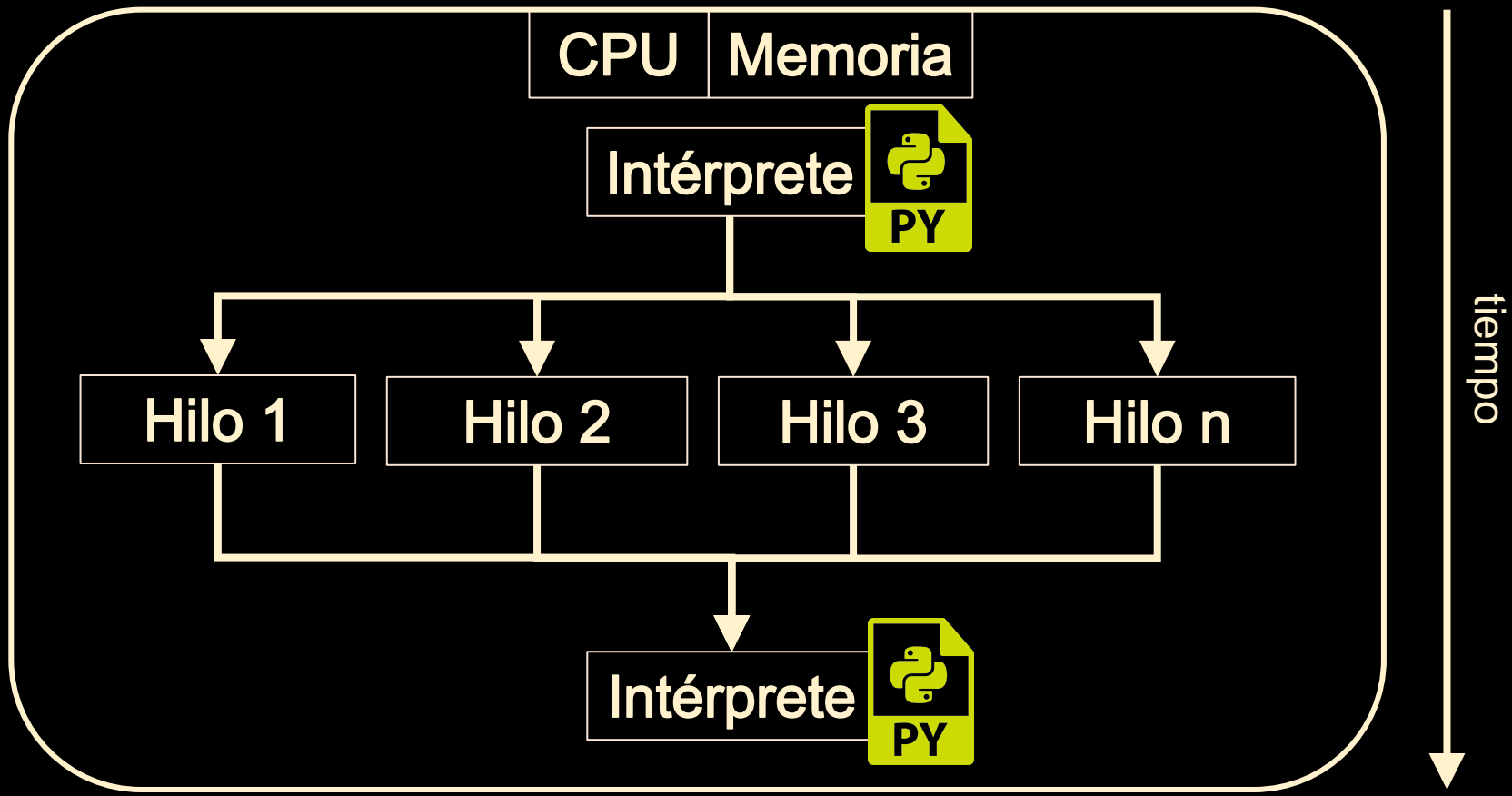
El traductor trabado

- Lenguaje anterior computación multi-hilos
- **Problema**: manejo no seguro de la memoria
- **Solución**: *Global Interpreter Lock* (GIL), bloqueo: interpretación en un solo hilo
- Python por naturaleza es no paralelo
- Nota: julia sí, taller? 

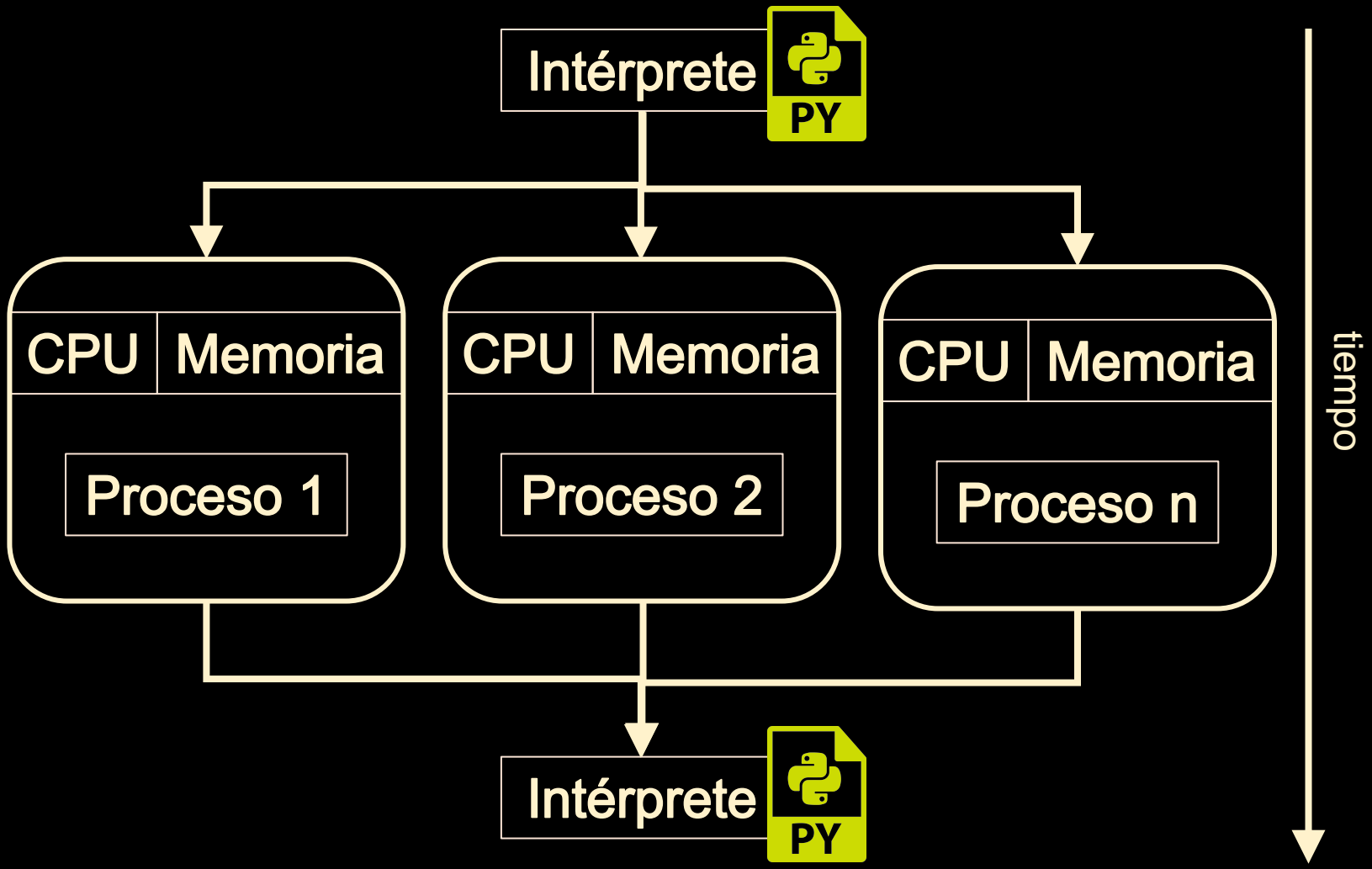
¿Como lidia python con este GIL?

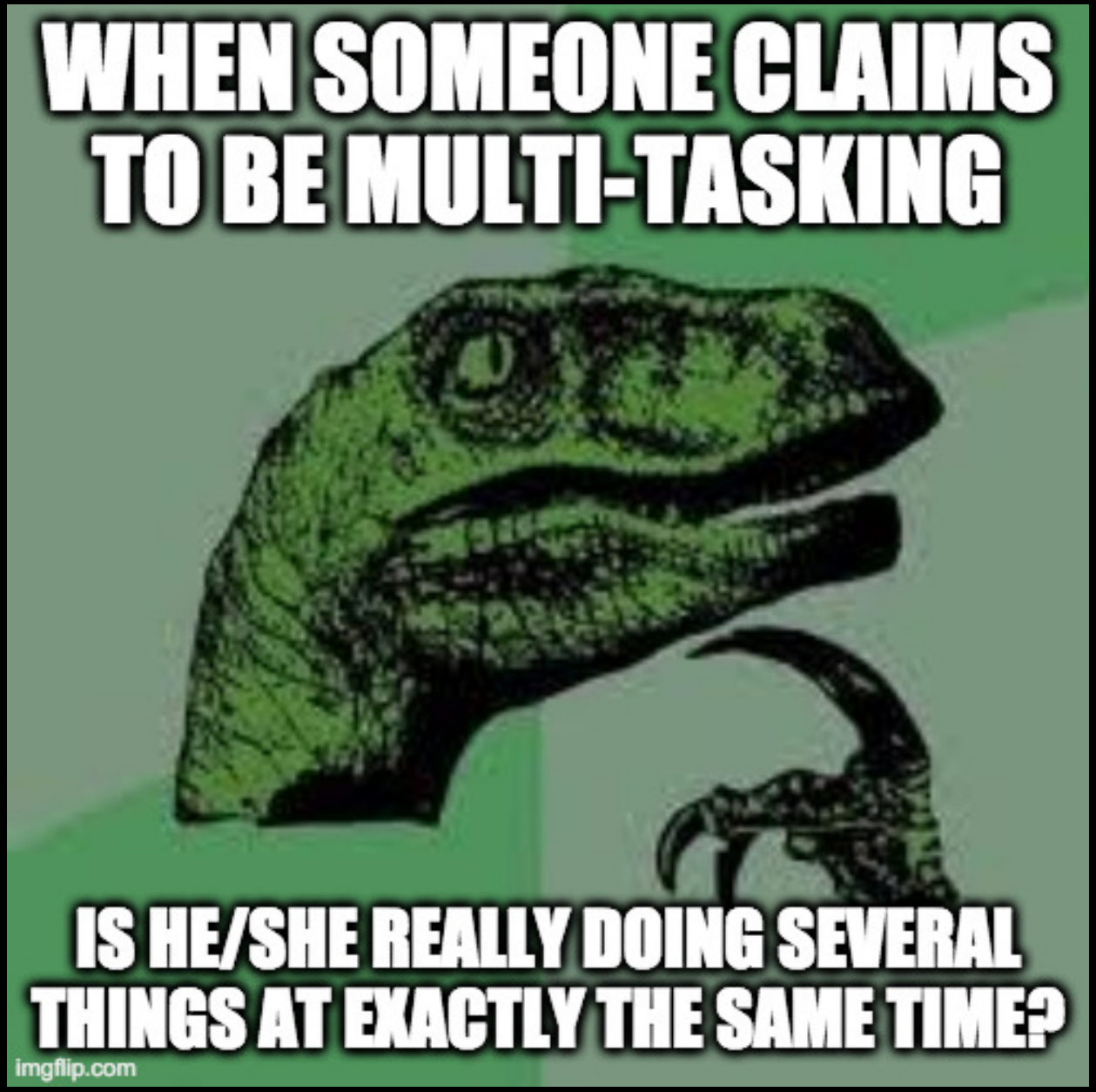


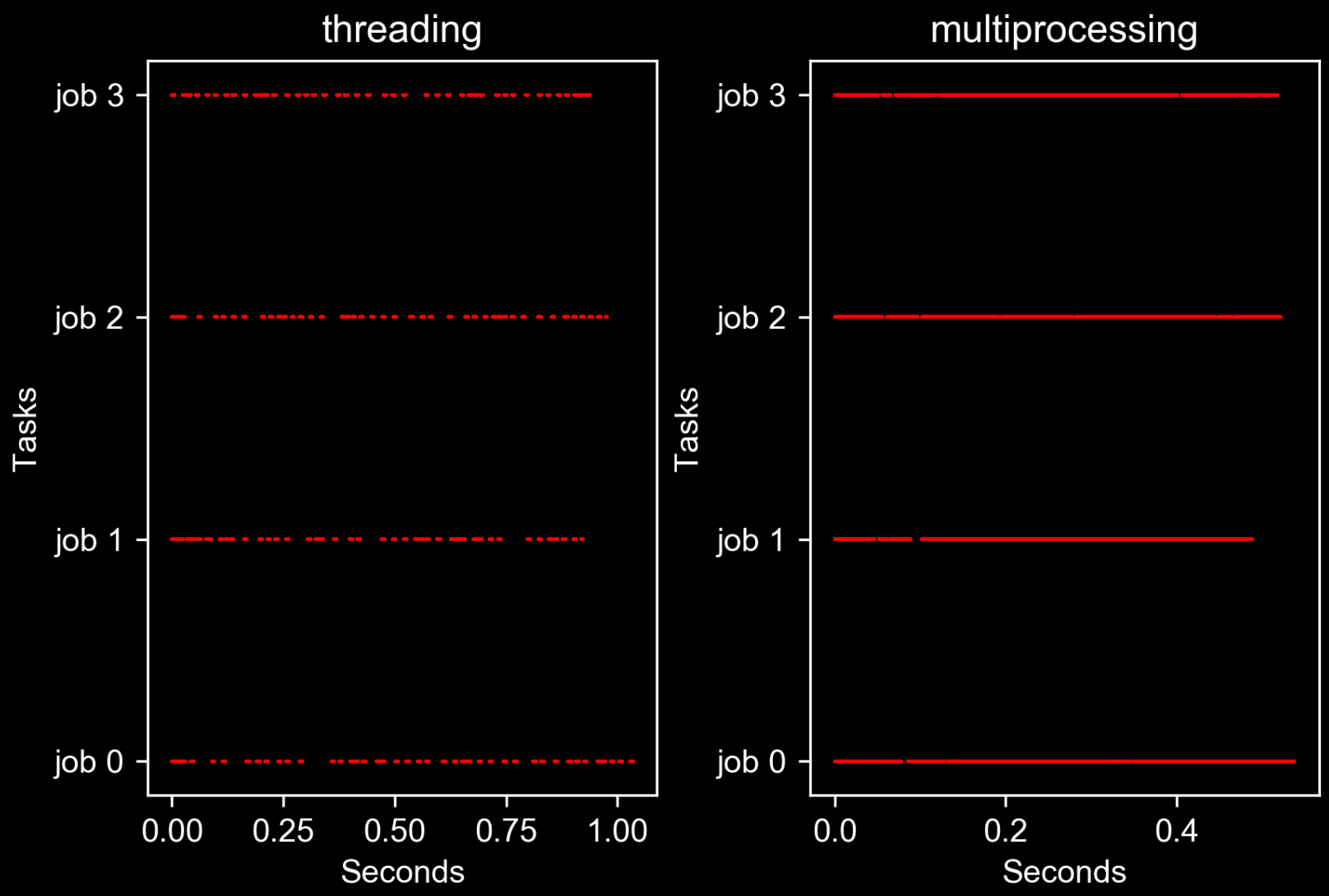
threading


CPU
Memoria
Intérprete
PY
Hilo 1
Hilo 2
Hilo 3
Hilo n
Intérprete
PY
tiempo

multiprocessing




WHEN SOMEONE CLAIMS TO BE MULTI-TASKING
IS HE/SHE REALLY DOING SEVERAL THINGS AT EXACTLY THE SAME TIME?
imgflip.com



https://github.com/baatout/multithreading-vs-multiprocessing.git

`threading`

- Hilos
- Memoria compartida
- Arranque ligero
- I/O, network, DB, GUI

`multiprocessing`

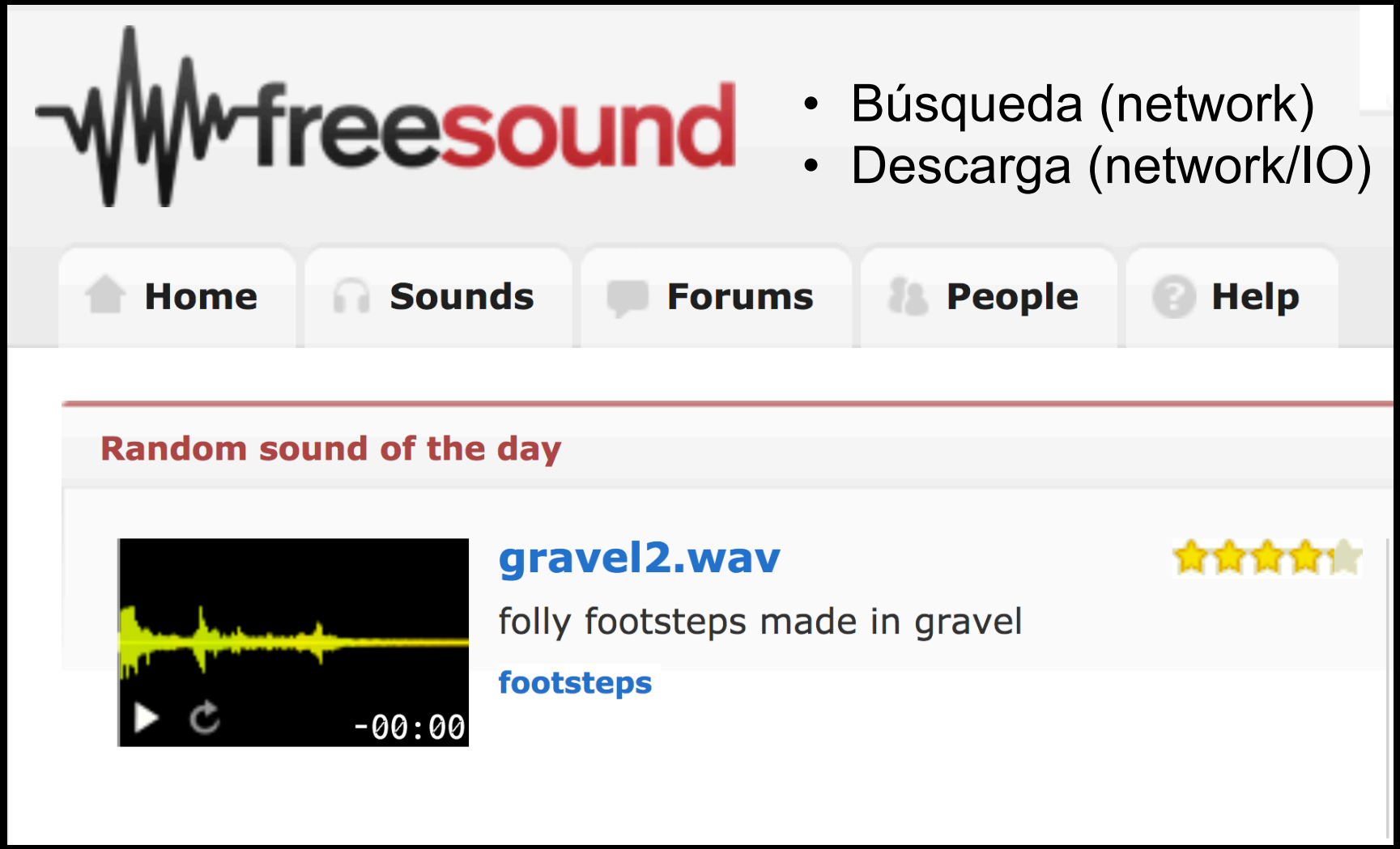
- Procesos
- Independiente
- Arranque pesado
- CPU’s

Sintáxis muy parecida

Ejemplo: `threading`

Web scraping

Ejemplo 0: raspar sonido libre



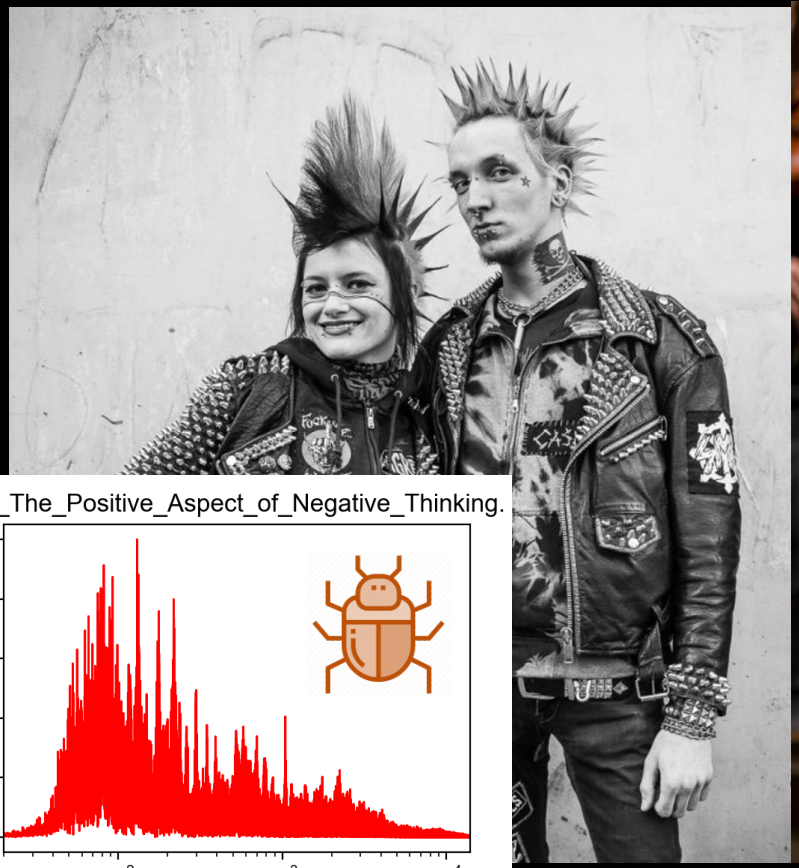
- Búsqueda (network)
- Descarga (network/IO)

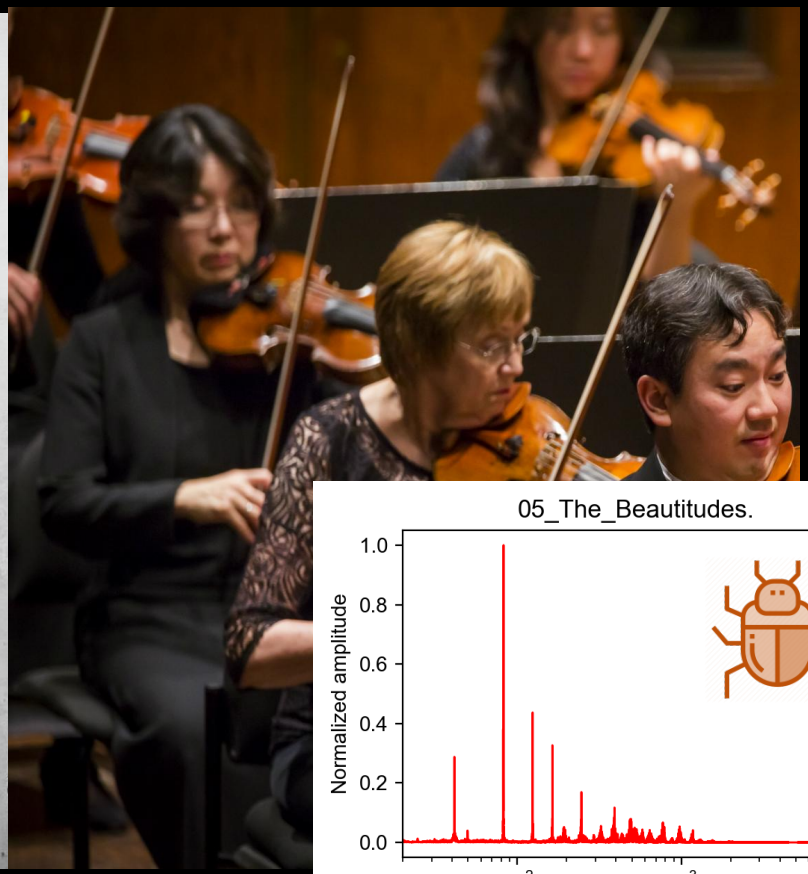
Ejemplo 1

multiprocessing, pymp, y mpi4py

Objetivo: Comparación espectral de géneros musicales

- mp3 a wav
- FFT
- Graficar



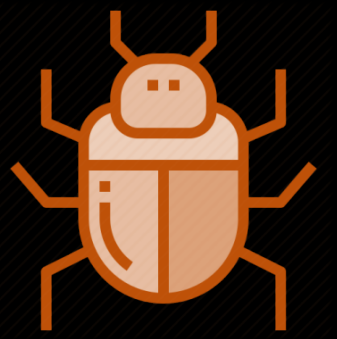


Ejemplo 1: `multiprocessing`

mp3 → wav

Ejemplo 2: `multiprocessing`

Comparación espectral de géneros musicales



Ejemplo 2: PyMP

Comparación espectral de géneros musicales

El *patois* con MPI

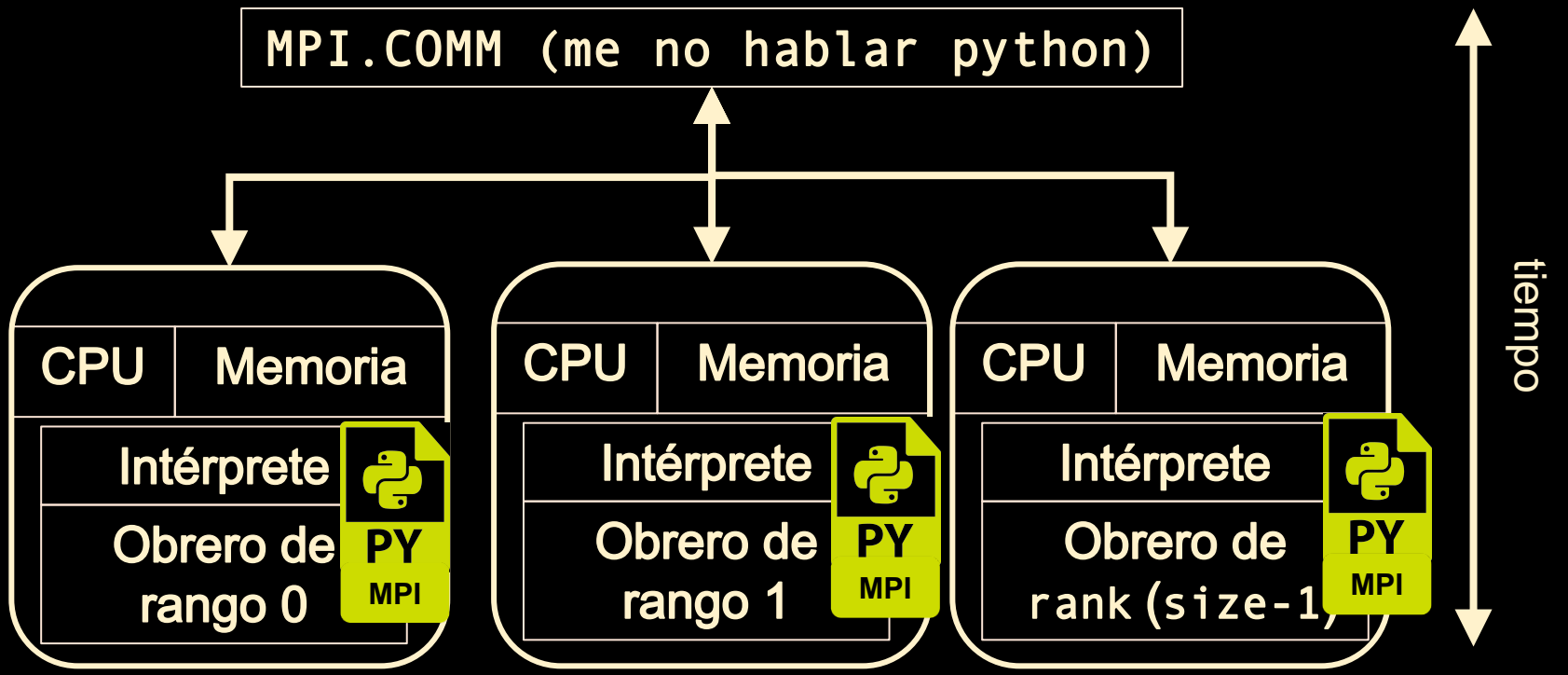
`mpi4py`

MPI: Interface de Paso de Mensajes

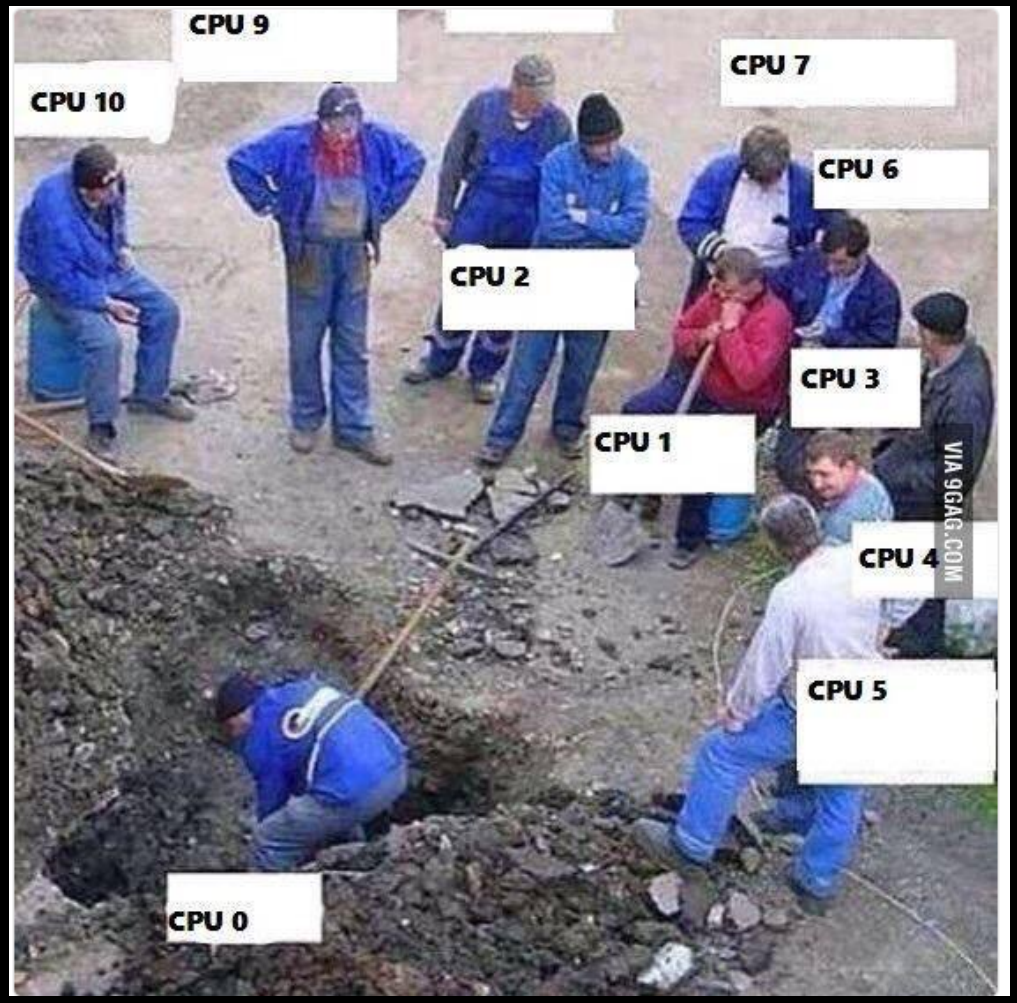
MPI: Requerimientos

- Instalación de MPI en MPICH o OpenMP
- Instalación de biblioteca MPI4Py
  - `pip install mpi4py`
  - `conda install -c anaconda mpi4py`

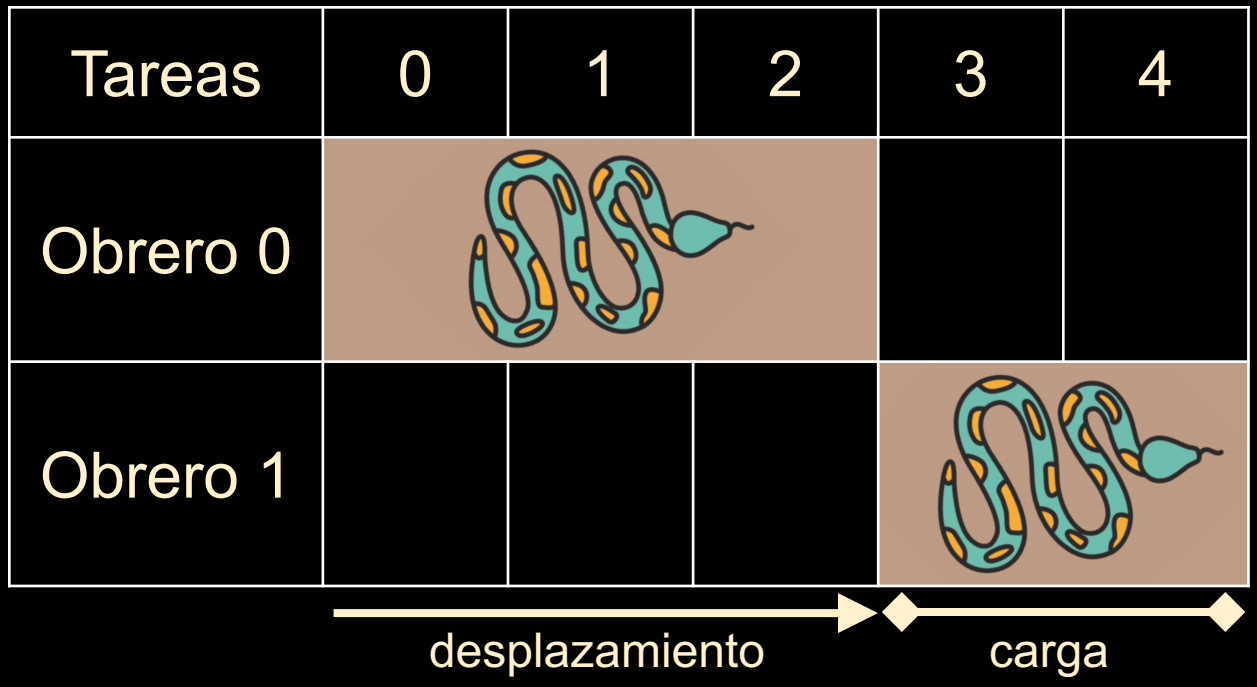
MPI


MPI.COMM (me no hablar python)
CPU
Memoria
Intérprete
Obrero de rango 0
PY
MPI
CPU
Memoria
Intérprete
Obrero de rango 1
PY
MPI
CPU
Memoria
Intérprete
Obrero de rank (size-1)
PY
MPI
tiempo

Repartición de tareas en MPI



Repartición de tareas en MPI



rank	displs	counts
0	0	3
1	3	2

Ejemplo 2: `mpi4py`

Comparación espectral de géneros musical

Recapitulación del repaso

Necesidad	Método
Net/IO	`threading`
Desarrollo rápido	`multiprocessing`
Desarrollo rápido - código, - control	`pymp`
• Control • Transparencia • Multinodos	`mpi4py`

```
leo:~/parallel_python_workshop$ mpiexec python gracias.py
¡Muchas gracias!
¡Muchas gracias!
¿Preguntas, comentarios?
¿Preguntas, comentarios?
```

lvanderlaat/parallel_python_workshop