{Python multithreading}

Python Catania - Stefano Borzì

```
from time import sleep

arr = range(100)

for x in arr:
    print(x)
    sleep(1)
```

tqdm

$ pip install tqdm

```python
from time import sleep
from tqdm import tqdm

arr = range(100)

for x in tqdm(arr):
    # print(x)
    sleep(1)
```

```
python example.py
 8%|▉                                        | 8/100 [00:08<01:32,  1.00s/it]
```

https://github.com/tqdm/tqdm

pathos

$ pip install pathos

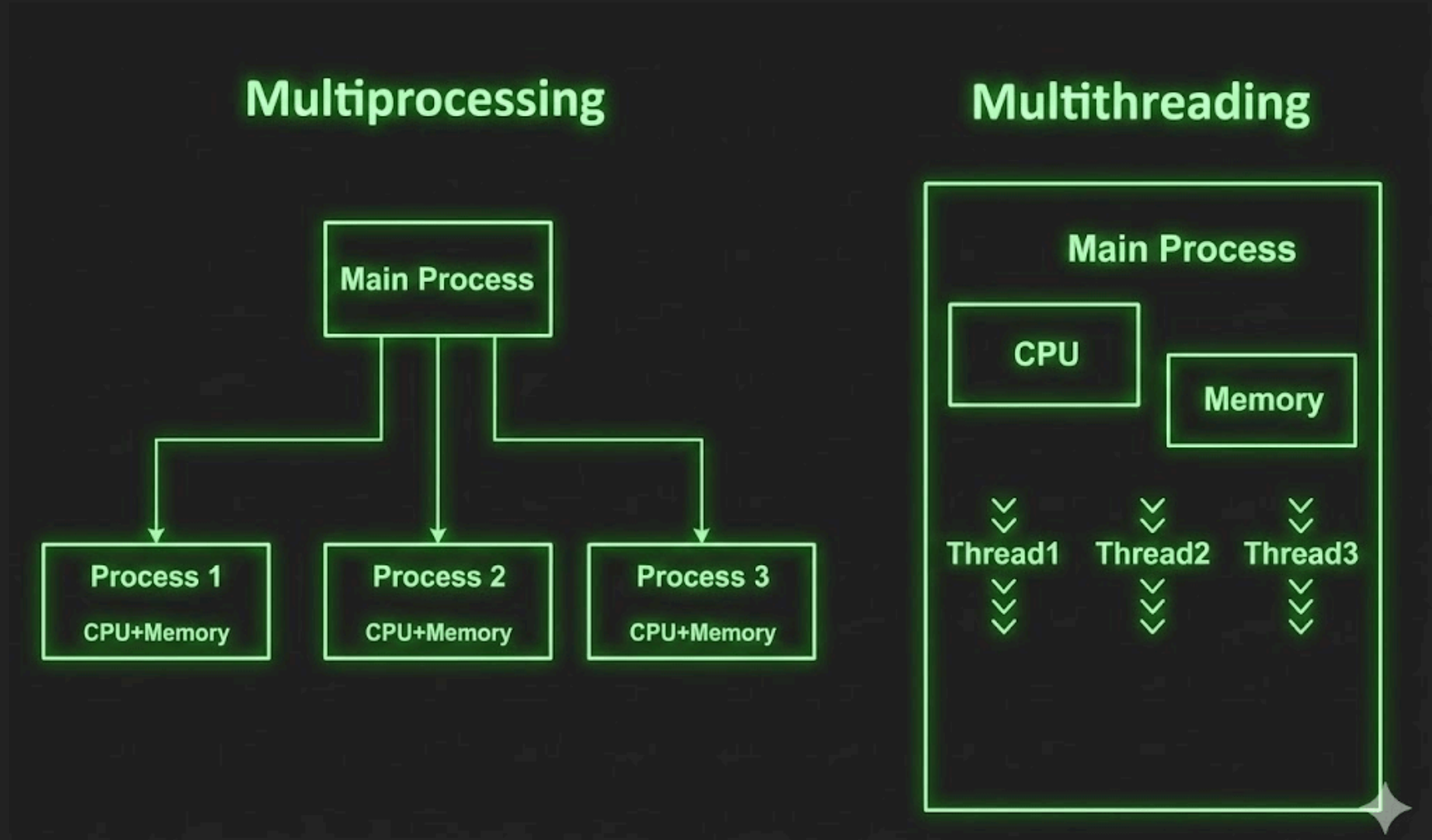
```
from time import sleep
import pathos as pa

arr = range(100)

def parallel():
    sleep(1)

ncpu = pa.helpers.cpu_count()
with pa.multiprocessing.ProcessingPool(ncpu) as p:
    p.map(parallel, arr), total=len(arr)
```

https://github.com/uqfoundation/pathos

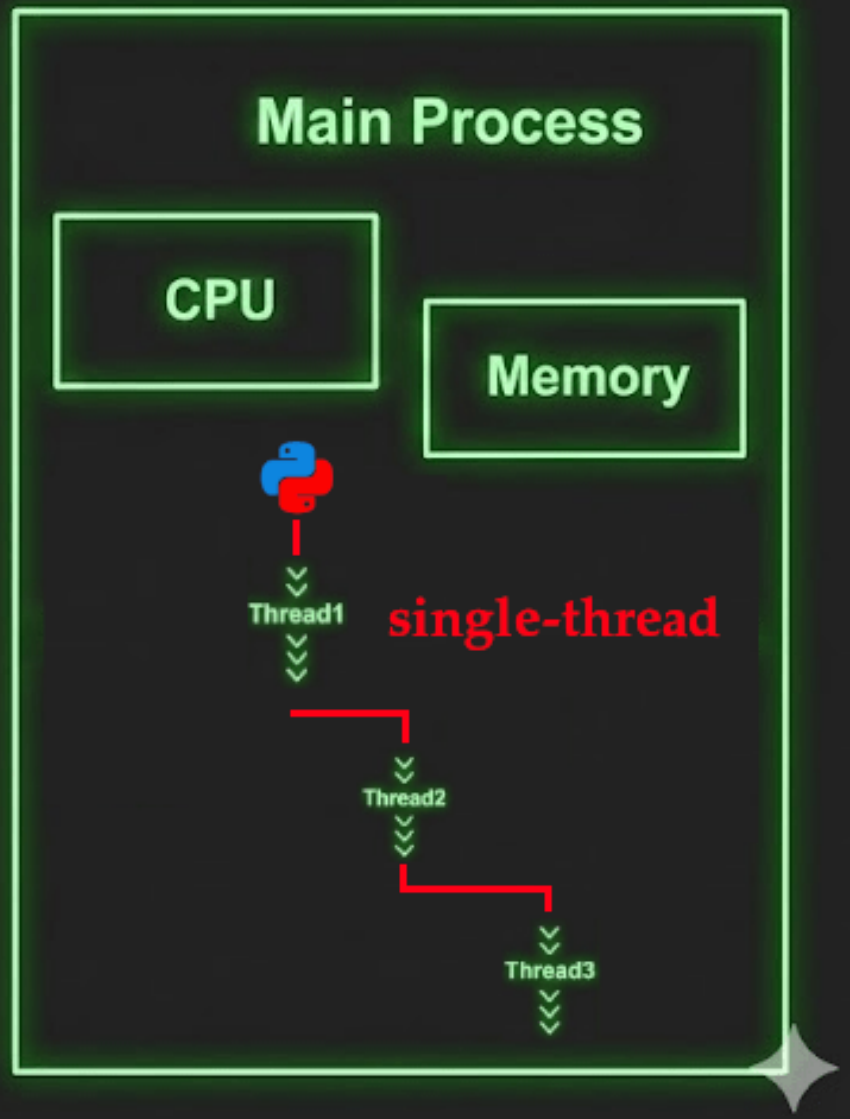
pathos + tqdm

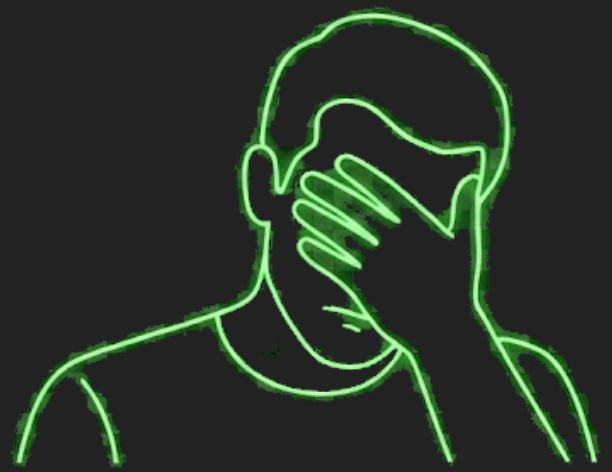
```
from time import sleep
from tqdm import tqdm
import pathos as pa

arr = range(100)

def parallel():
    sleep(1)

ncpu = pa.helpers.cpu_count()
with pa.multiprocessing.ProcessingPool(ncpu) as p:
    list(tqdm(p.imap(parallel, arr), total=len(arr)))
```

multiprocessing vs multithreading

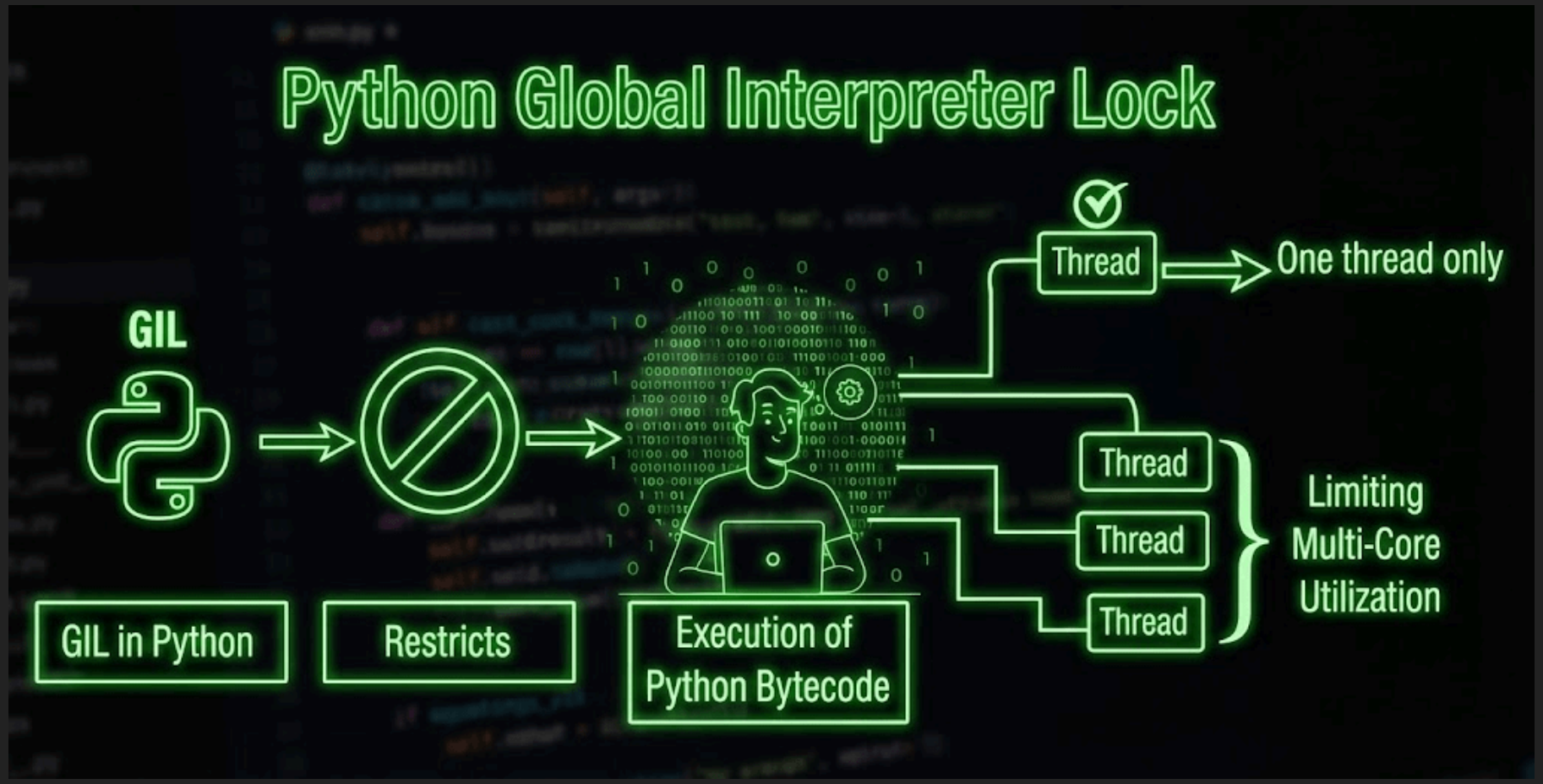


multithreading in Python 3.12 (single-thread)



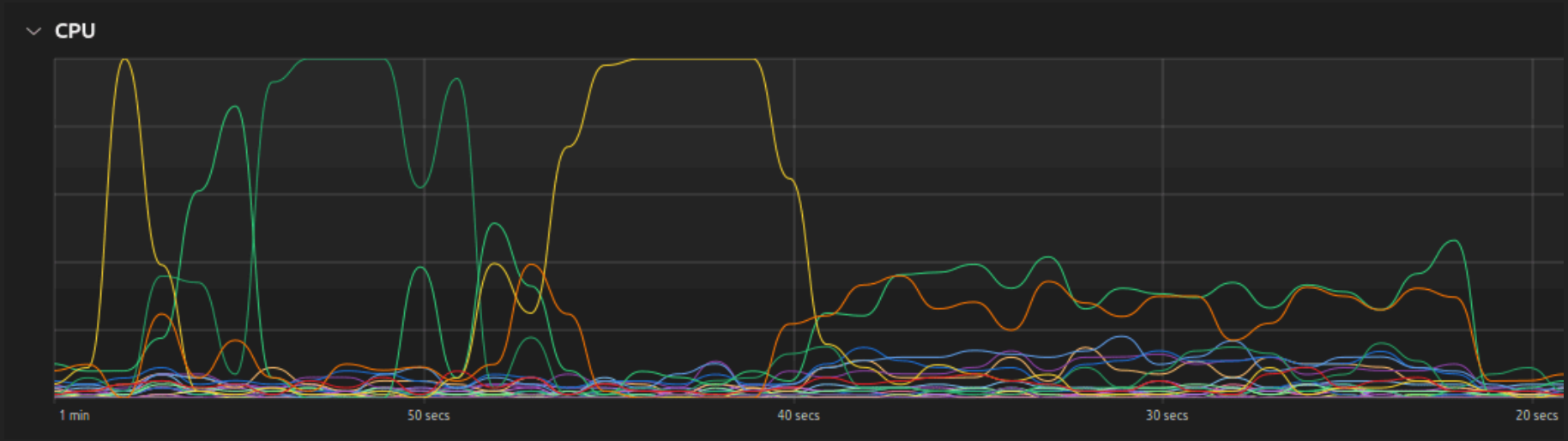


multithreading - Python GIL



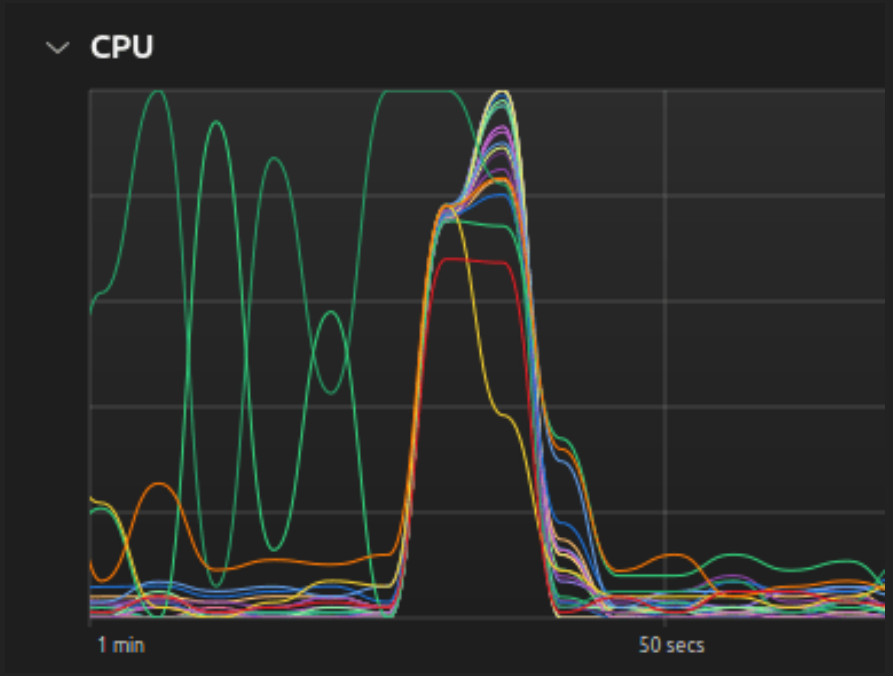
Python has a built-in mechanism that limits how threads are executed, and it’s called the Global Interpreter Lock, or GIL.

CPU simulation - single thread (gil)

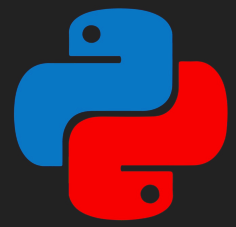



```
--- Python Version: 3.12.3 ---
--- GIL Status: Active (Standard) ---


--- CPU Cores Available: 22 ---
--- Simulating 22 Threads ---

Running Sequential Test... Done! (21.4386s)
Running Threaded Test...   Done! (19.5765s)

========================================
Speedup Factor: 1.10x
RESULT: FAIL. No significant speedup.
The GIL prevented threads from running in parallel.
========================================
```

CPU simulation - multithreading (no-gil)




```
--- Python Version: 3.14.2 ---
--- GIL Status: Disabled (Free-Threaded) ---


--- CPU Cores Available: 22 ---
--- Simulating 22 Threads ---

Running Sequential Test... Done! (19.6890s)
Running Threaded Test...   Done! (2.6912s)

========================================
Speedup Factor: 7.32x
RESULT: SUCCESS. Effective Parallelism.
Your code ran 7.32 times faster using threads.
========================================
```

exit(0) 

Thank you!