

I/O-Bound vs CPU-Bound Work in Python

Why this distinction exists

Choosing the wrong concurrency model leads to slow and fragile systems. The first decision is not threads or async, but whether the program is waiting on I/O or on the CPU.

Target Usage (I/O-bound)

```
import time

def fetch():
    time.sleep(1)  # waiting on I/O
```

Target Usage (CPU-bound)

```
def compute():
    total = 0
    for i in range(10_000_000):
        total += i * i
    return total
```

Coding Problem

Decide whether concurrency will help a workload and which model to use, based on whether the program is primarily waiting or computing.

Observed Effect

Threads and async improve I/O-bound workloads by overlapping waiting. CPU-bound workloads do not benefit from threads due to the GIL.

Key Insight

Concurrency overlaps waiting; parallelism speeds up computation. Correct classification must happen before choosing a concurrency model.