Course / Lesson 7: Concurrency & Async Programming / Lesson 7 Content

Previous ✓ ✓ Next

Part 2: Other Alternatives

Bookmark this page

Part 2: Other Alternatives

Traditionally, concurrency has been achieved through multiple process communication and in-process threads, as we've seen.

Another strategy is through micro-threads, implemented via coroutines and a scheduler.

A coroutine is a generalization of a subroutine which allows multiple entry points for suspending and resuming execution.

The threading and the multiprocessing modules follow a preemptive multitasking model

Coroutine based solutions follow a cooperative multitasking model:

Threads versus processes in Python

Threads are lightweight processes, run in the address space of an OS process, true OS level threads.

Therefore, a component of a process.

This allows multiple threads access to data in the same scope.

Threads can not gain the performance advantage of multiple processors due to the Global Interpreter Lock (GIL).

But the GIL is released during IO, allowing IO bound processes to benefit from threading.

Processes

A process contains all the instructions and data required to execute independently, so processes do not share data!

Multiple processes are best to speed up CPU bound operations.

The Python interpreter isn't lightweight!

Communication between processes can be achieved via:

`multiprocessing.Queue>`

`multiprocessing.Pipe<`

and regular IPC (inter-process communication)

Data moved between processes must be pickleable (it can be serialized).

Advantages / Disadvantages of Threads

Advantages:

They share memory space:

- Threads are relatively lightweight – shared memory means they can be created fairly quickly without much memory use.
- Easy and cheap to pass data around (you are only passing a reference).

Disadvantages:

They share memory space:

- Each thread is working with the *same* python objects.
- Operations often take several steps and may be interrupted mid-stream
- Thus, access to shared data is also non-deterministic

(race conditions)

‹ Previous

Next ›