

CODAY - ADDE SISTEMAS

ENTENDENDO THREADS, PROCESSOS E ASYNCIO COM PYTHON

Apresentado por Anderson Rocha

VAMOS USAR O SEGUINTE EXEMPLO:

IMAGINE QUE VOCÊ
VAI FAZER UM
JANTAR PARA 10
AMIGOS.

Os passos para fazer o prato proposto para a janta são os seguintes:

- Lavar os vegetais
- Cortar os vegetais
- Cozinhar
- Servir

Para cada passo acima, você leva os seguintes tempos:

- 5 minutos (Lavar os vegetais)
- 13 minutos (Cortar os vegetais)
- 30 minutos (Cozinhar)
- 2 minutos (Servir)

A SOMA DOS PASSOS É: 50 MINUTOS. LOGO, PARA 10 AMIGOS, VOCÊ DEMORARIA 500 MINUTOS, UM POUCO MAIS DE 8 HORAS.

ACREDITO QUE ESSE TEMPO POSSA SER REDUZIDO!

PARALELISMO

Imagine que você quer reduzir o tempo de cozimento para 250 minutos.

Se algum amigo te ajudar a cozinhar, logo, vocês executarão o serviço em paralelo e irão reduzir o tempo para 250 minutos.

O que acontece nesse caso, é que cada pessoa está "cozinhando na sua própria cozinha" ou seja, elas não estão disputando recursos em comum.

Ou seja, paralelismo acontece quando o computador tem vários cores. E existem processos rodando em ao menos dois deles.

O paralelismo, acontece através da execução de diferentes processos.

CONCURRENCY / CONCORRÊNCIA / SIMULTANEIDADE

Podemos ver que o nosso maior gargalo acontece quando vamos cozinhar é a tarefa de levar os legumes para o forno.

Uma pergunta que podemos fazer é a seguinte: quantos pratos posso cozinhar por vez no forno? Se pudéssemos cozinhar os 10 pratos de uma só vez, vamos economizar 270 minutos.

Nesse caso, não vamos utilizar recurso algum de outra cozinha, simplesmente aproveitamos da capacidade de um recurso ocioso.

CONCURRENCY / CONCORRÊNCIA / SIMULTANEIDADE

A concorrência acontece dentro do mesmo processo, aproveitando ociosidade de um recurso.

Poderíamos aplicar facilmente concorrência às outras etapas também.

Concorrência acontece dentro de uma thread que está sendo executada dentro de um processo. Um processo pode ter várias threads. As threads podem compartilhar recursos entre elas, por exemplo o forno.

QUANDO USAR UM OU OUTRO?

Falando por cima, se sua aplicação Python está executando operações vinculadas à CPU, como processamento de números ou manipulação de texto, opte pelo paralelismo. A simultaneidade não trará muitos benefícios nesses cenários.

Paralelismo -> CPU Bound

Concorrência -> IO Bound

REFERÊNCIAS

<https://medium.com/fintechexplained/advanced-python-concurrency-and-parallelism-82e378f26ced>

<https://danielflannery.ie/simulate-cpu-load-with-python/>

<https://api.openbrewerydb.org/>

<https://github.com/public-apis/public-apis>

Let's ~~code~~ coday

YOU ROCK

Thanks

