链滴

Python 线程池 threadpool

作者：shiguofu

原文链接：https://ld246.com/article/1544683394424

来源网站：链滴

最近在试用python抓取一些新闻，用到了多线程模块，记录下来分享下；

不使用多进程是觉得多进程占用资源，爬虫模块限制占用资源，力求快速，因此使用了python的线池

python多线程

多线程可以使程序同时执行多个事件，提高CPU的利用率，节省多个任务的执行事件；

在python中，主要用

1. multiprocessing.pool 多进程多线程
2. threadpool 多线程

这两个模块来实现线程池

线程池threadpool

线程池与线程的区别就是，线程池是提前创建固定的线程数量，超过线程数量就会等待；而线程是根需求创建线程，创建的数量是由系统决定的；

一、安装

安装python-pip，然后使用pip安装；安装后测试是否安装成功

```
pip install threadpool
python -c "import threadpool"       # 测试是否安装成功，无输出则成功，否则失败
```

二、代码实例

Talk is cheap, show me the code.

先上一段代码，自己体会；

```
import time
import threadpool

def worker(params1，  params2):
  time.sleep(2)
  print("hello world", params1, params2)


def test_pool():
  pool = threadpool.ThreadPool(5)
  nums_tasks = 10
  for i in range(nums_tasks):
    # requests = threadpool.makeRequests(worker, [([i, i + 1], {})])  # 参数传递,遵循python参解包
    requests = threadpool.makeRequests(worker, [([], {"params1": i, "params2": i + 1})])
    pool.putRequest(requests)
  pool.wait()
```

原文链接：Python 线程池 threadpool

```
if __name__ == '__main__':
    test_pool()
```

在test_pool中，首先创建一个拥有5个线程的线程池，在for循环中，通过threadpool.makeRequest，构建线程池请求任务，特别注意参数传递的方式；

threadpool.makeRequests的参数是一个tuple，第一个为list，依次对应参数的第一个、第二个...；二个为dict，传参key=value的格式；不可重复传递参数，也不能少传递参数；

最后通过pool.putRequest将任务丢到线程池执行；pool.wait 等待所有线程结束；

多线程 multiprocessing

multiprocessing是python的标准库，多进程，多线程都支持

代码实例

```
import time
from multiprocessing.pool import ThreadPool

def worker(params1，  params2):
   time.sleep(2)
   print("hello world", params1, params2)


def test_pool():
   pool = ThreadPool(5)
   nums_tasks = 10
   for i in range(nums_tasks):
      pool.apply_async(worker, (i, i + 1))
   pool.close()
   pool.join()


if __name__ == '__main__':
    test_pool()
```

multiprocessing的使用要简单一点，只需要创建线程池，执行线程，传递参数；

最后都需要调用pool.close才可以join，否则报错 ---- 疑问?

执行上述代码，发现比串行快了许多；达到最终效果；