

线程池和进程池

当我们对某些网站内容进行抓取的时候非常容易遇到这样一种情况。

The screenshot shows the website www.xinfadi.com.cn displaying a list of market prices for various goods. The URL in the browser address bar is `http://www.xinfadi.com.cn/marketanalysis/0/list/14293.shtml`. The page shows a table of prices with columns for item name, lowest price, average price, highest price, specification, unit, and release date. The pagination at the bottom indicates there are 285849 items in total, and the current page is 14293. A red arrow highlights the connection between the URL and the page number.

品名	最低价	平均价	最高价	规格	单位	发布日期
白蛤	3.50	4.00	4.50	活	斤	2019-01-01
青蛤	12.00	12.50	13.00	活	斤	2019-01-01
花蛤	4.50	4.75	5.00	活	斤	2019-01-01
毛蚶	3.00	3.50	4.00	活/南方	斤	2019-01-01
毛蚶	4.00	4.50	5.00	活/北方	斤	2019-01-01
仿野生甲鱼	40.00	41.50	43.00	活	斤	2019-01-01
养殖甲鱼	25.00	26.50	28.00	活	斤	2019-01-01
蚕蛹	33.00	33.50	34.00	活	斤	2019-01-01
牛蛙	7.00	7.50	8.00	活	斤	2019-01-01

看这个网站, 我们发现这网站的数据太多了. 有一万多页. 也就对应着一万多个url. 那我们设计多线程的时候如果每个url对应一个线程就会产生新问题. 朋友, 你一定要知道. 创建线程本身也是要消耗你的计算机资源的. 线程不是变魔术变出来的. 那这时我们就可以考虑能不能重复的使用线程呢? 答案当然可以. 线程池就可以帮你搞定.

线程池工作原理:

创建一个大池子, 存放固定数量的线程. 然后把我们要执行的任务丢给线程池. 由线程池去分配哪个线程来完成该任务. 其他的事情都不需要你来管. 舒服吧.

废话不多说, 上代码

```
1 from concurrent.futures import
  ThreadPoolExecutor, ProcessPoolExecutor
2
3 # 线程池
4 def fn(name):
5     for i in range(1000):
6         print(name, i)
7
8
9 if __name__ == '__main__':
10     with ThreadPoolExecutor(10) as t:
11         for i in range(100):
12             t.submit(fn, name=f"线程{i}")
13
```

至于进程池. 就把ThreadPoolExecutor更换为ProcessPoolExecutor就可以了. 其他一模一样