

python 多进程与多线程获取股票数据


前一天简单介绍了 python 怎样获取历史数据和实时分笔数据，那么如果要获取所有上市公司的实时分笔数据，应该怎么做呢？

肯定有人想的是，用一个列表存储所有上市公司的股票代码，然后无限循环获取不就得了吗？

现在深市和沪市的股票一共有 3400 多只，如果你真这样做的话，获取一次所有股票的实时数据需要十几二十秒的时间，甚至更多，而且非常容易因为等待超时而使程序挂掉，如果你的模型对实时数据的质量要求非常高，这肯定是不行的，即使不考虑数据质量，获取数据的脚本经常挂也是不行的。

那首先想到的是多线程和多进程。然而因为 python 全局解释锁的存在，在多线程情况下，即使是在多核 cpu 的情况下，也只能同时执行一个线程，即使如此，多线程获取一次所有股票实时数据所花的时间，依然比在一个大循环里面运用单线程少得多，这是因为 python 从网络获取数据存储到本地，是 IO 密集型任务，python 多线程依然能很大程度上提高性能，具体细节在这里不多做介绍。

为了充分利用带宽资源，IO 资源，在这里使用多线程和多进程两种方式获取股票数据，首先我们需要有一个比较全的所有股票代码文件，我已经替大家准备好了，关注微信公众号【数据之佳】回复“股票”四个字即可得到下载链接，压缩包提供多线程和多进程获取股票实时数据两个案例，其中的 stocks 文件内存储的是股票代码文件，提供的是截止今日（2017-11-23）的所有上市公司代号。

名称	修改日期	类型	大小
 get_realtime_quotes_with_multiprocess.py	2017/11/23 21:48	Python File	1 KB
 get_realtime_quotes_with_multithread.py	2017/11/23 21:51	Python File	1 KB
 stocks	2017/11/23 21:00	文件	24 KB

在这里提供多线程的例子，多进程的例子请在【数据之佳】里面查看，同时代码也给大家准备好了，上面回复的“股票”得到的下载链接里面直接就有 python 源文件，下载以后修改一下路径直接就可以跑了，其中用到的 python 版

本是昨天分享的文章中 anaconda 自带的 python3.5, 代码依然使用 jupyter 编写, 操作系统为 win10, 在 linux 上只需要稍加改动就可以执行了。

```
import pandas as pd
import numpy as np
import tushare as ts
import os
import time
from threading import Thread
def get_data(stock):
    date=time.strftime('%Y-%m-%d-%H-%M-%S',time.localtime(time.time()))
    data=ts.get_realtime_quotes(stock)
    print(data)
    path='F:\\stocks'
    data.to_csv(path+'\\'+date+'_'+stock,mode='wt')
time1=time.strftime('%Y-%m-%d-%H-%M-%S')
print(time1)
with open('F:\\stocks\\stock_codes\\stocks') as f:
    try:
        while True:
            line=next(f).strip()
            t=Thread(target=get_data,args=(line,))
            t.start()
    except StopIteration:
        pass
time2=time.strftime('%Y-%m-%d-%H-%M-%S')
print(time2)
```

其中 time1 是获取一次所有股票的开始时间, time2 是结束时间, 具体有耗时多上时间, 请读者下载元代以后自行测试

2017-11-23-21-16-23
2017-11-23-21-16-33

这是我注释了 `print(data)` 一行以后所花的时间, 恰好用了 10 秒, 如果你的网络好一些, 可能会更快, 当然这不是最快的方法, 最快的方法可以实现一秒

以内获取一次。

stock_codes	2017/11/23 21:01	文件夹	
2017-11-23-21-06-49_000058	2017/11/23 21:06	文件	1 KB
2017-11-23-21-06-49_000420	2017/11/23 21:06	文件	1 KB
2017-11-23-21-06-49_000901	2017/11/23 21:06	文件	1 KB
2017-11-23-21-06-49_002049	2017/11/23 21:06	文件	1 KB
2017-11-23-21-06-49_002102	2017/11/23 21:06	文件	1 KB
2017-11-23-21-06-49_002119	2017/11/23 21:06	文件	1 KB
2017-11-23-21-06-49_002214	2017/11/23 21:06	文件	1 KB
2017-11-23-21-06-49_002239	2017/11/23 21:06	文件	1 KB
2017-11-23-21-06-49_002245	2017/11/23 21:06	文件	1 KB
2017-11-23-21-06-49_002261	2017/11/23 21:06	文件	1 KB
2017-11-23-21-06-49_002323	2017/11/23 21:06	文件	1 KB
2017-11-23-21-06-49_002407	2017/11/23 21:06	文件	1 KB
2017-11-23-21-06-49_002478	2017/11/23 21:06	文件	1 KB
2017-11-23-21-06-49_002497	2017/11/23 21:06	文件	1 KB
2017-11-23-21-06-49_002520	2017/11/23 21:06	文件	1 KB
2017-11-23-21-06-49_002670	2017/11/23 21:06	文件	1 KB
2017-11-23-21-06-49_002690	2017/11/23 21:06	文件	1 KB
2017-11-23-21-06-49_002777	2017/11/23 21:06	文件	1 KB
2017-11-23-21-06-49_002843	2017/11/23 21:06	文件	1 KB

数据已经存储在指定目录下，使用多进程的例子请各位自行获取下载链接下载。

下面是【数据之佳】的二维码，我会在公众号分享一些量化建模的案例，结果等。

