# 四号程序员

Keep It Simple And Stupid

# Python多进程并发(multiprocessing)

由于Python设计的限制(我说的是咱们常用的CPython)。最多只能用满1个CPU核心。

Python提供了非常好用的多进程包multiprocessing,你只需要定义一个函数,Python会替你完成其他所有事情。借助这个包,可以轻松完成从单进程到并发执行的转换。

1、新建单一进程

如果我们新建少量进程,可以如下:

```
import multiprocessing
   import time
2
3
   def func(msg):
5
       for i in xrange(3):
6
            print msg
7
           time.sleep(1)
8
9
   if __name__ == "__main__":
       p = multiprocessing.Process(target=func, args=("hello", ))
10
11
       p.start()
12
       p.join()
       print "Sub-process done."
13
```

### 2、使用进程池

是的,你没有看错,不是线程池。它可以让你跑满多核CPU,而且使用方法非常简单。

注意要用apply\_async,如果落下async,就变成阻塞版本了。

processes=4是最多并发进程数量。

```
1 import multiprocessing
2 import time
3
4 def func(msg):
5   for i in xrange(3):
6     print msg
```

```
7
           time.sleep(1)
8
  if __name__ == "__main__":
9
10
       pool = multiprocessing.Pool(processes=4)
       for i in xrange(10):
11
           msg = "hello %d" %(i)
12
13
            pool.apply_async(func, (msg, ))
       pool.close()
14
15
       pool.join()
       print "Sub-process(es) done."
16
```

#### 3、使用Pool,并需要关注结果

更多的时候,我们不仅需要多进程执行,还需要关注每个进程的执行结果,如下:

```
import multiprocessing
2
   import time
3
4
   def func(msg):
5
       for i in xrange(3):
6
           print msg
7
            time.sleep(1)
8
       return "done " + msq
9
10 if __name__ == "__main__":
       pool = multiprocessing.Pool(processes=4)
11
12
       result = \Pi
13
       for i in xrange(10):
           msg = "hello %d" %(i)
14
            result.append(pool.apply_async(func, (msg, )))
15
16
       pool.close()
17
       pool.join()
18
       for res in result:
19
            print res.get()
       print "Sub-process(es) done."
20
```

#### 2014.12.25更新

根据网友评论中的反馈,在Windows下运行有可能崩溃(开启了一大堆新窗口、进程),可以通过如下调用来解决:

```
Python

1 multiprocessing.freeze_support()
```

### 您可能也喜欢如下文章:

- 1. 关于gevent的微线程"eventlet"同步问题
- 2. Python的sched模块和Timer类
- 3. Hamsterdb tesing code (python)
- 4. Python Essential Reference 4th 第1章 读书笔记
- 5. 关于gevent的Timeout(超时)问题......

This entry was posted in Python and tagged multiprocessing, Python, 多进程, 并发 on 2012-06-05 [https://www.coder4.com/archives/3352] .

## 13 thoughts on "Python多进程并发(multiprocessing)"



很有帮助啊!谢谢。



为什么一运行直接跑了一堆进程, 服务器都崩了



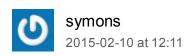
win32平台 增加 multiprocessing.freeze\_support()



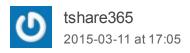
点赞!



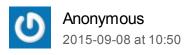
## 加了multiprocessing.freeze\_support()还是会那样报错,我用的win7x64



很有帮助,很给力!!!谢谢!!



真需要,写的不错 学习了!



werwerver2r3234234



multiprocessing.freeze\_support() 加在哪?



```
1 from multiprocessing import Process, freeze_support
2 ...
3 ...
4
5 if __name__ == '__main__':
6    freeze_support()
7 這樣加嗎?
```



是的



hi



加载什么文字freeze\_support()