

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：

```
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
9 if __name__ == "__main__":
10     p = multiprocessing.Process(target=func, args=("hello", ))
11     p.start()
12     p.join()
13     print "Sub-process done."
```

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

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：

```
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     return "done " + msg
9
10 if __name__ == "__main__":
11     pool = multiprocessing.Pool(processes=4)
12     result = []
13     for i in xrange(10):
14         msg = "hello %d" %(i)
15         result.append(pool.apply_async(func, (msg, )))
16     pool.close()
17     pool.join()
18     for res in result:
19         print res.get()
20     print "Sub-process(es) done."
```

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)”



Anonymous

2013-04-20 at 21:00

很有帮助啊！谢谢。



qyc

2014-12-21 at 10:20

为什么一运行直接跑了一堆进程，服务器都崩了



lly

2014-12-22 at 09:50

win32平台 增加 multiprocessing.freeze_support()



coder4

Post author

2014-12-25 at 18:25

点赞！



roots01

2015-08-28 at 09:36

加了 `multiprocessing.freeze_support()` 还是会那样报错,我用的 **win7x64**



symons

2015-02-10 at 12:11

很有帮助, 很给力!!! 谢谢!!



tshare365

2015-03-11 at 17:05

真需要, 写的不错 学习了!



Anonymous

2015-09-08 at 10:50

werwerwer2r3234234



請問

2015-09-17 at 09:00

`multiprocessing.freeze_support()` 加在哪?



請問

2015-09-17 at 09:15

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

**coder4**

Post author

2015-09-17 at 09:24

是的

**Anonymous**

2015-12-29 at 13:03

hi

**cavities@sina.com**

2016-02-03 at 14:21

加载什么文字freeze_support()