

Vamei

编程，数学，设计

[博客园](#)[首页](#)[订阅](#)[管理](#)[随笔-209](#) [文章-1](#) [评论-3802](#)

Python标准库11 多进程探索 (multiprocessing包)

作者: Vamei 出处: <http://www.cnblogs.com/vamei> 欢迎转载，也请保留这段声明。谢谢！

在初步了解Python多进程之后，我们可以继续探索multiprocessing包中更加高级的工具。这些工具可以让我们更加便利地实现多进程。

进程池

进程池 (Process Pool) 可以创建多个进程。这些进程就像是随时待命的士兵，准备执行任务 (程序)。一个进程池中可以容纳多个待命的士兵。



“三个进程的进程池”

比如下面的程序：



```
import multiprocessing as mul

def f(x):
    return x**2

pool = mul.Pool(5)
rel = pool.map(f, [1, 2, 3, 4, 5, 6, 7, 8, 9, 10])
print(rel)
```



我们创建了一个容许5个进程的进程池（Process Pool）。Pool运行的每个进程都执行f()函数。我们利用map()方法，将f()函数作用到表的每个元素上。这与built-in的map()函数类似，只是这里用5个进程并行处理。如果进程运行结束后，还有需要处理的元素，那么的进程会被用于重新运行f()函数。除了map()方法外，Pool还有下面的常用方法。

`apply_async(func, args)` 从进程池中取出一个进程执行func，args为func的参数。它将返回一个AsyncResult的对象，你可以对该对象调用get()方法以获得结果。

`close()` 进程池不再创建新的进程

`join()` wait进程池中的全部进程。必须对Pool先调用close()方法才能join。

练习

有下面一个文件download.txt。

```
www.sina.com.cn
www.163.com
www.iciba.com
www.cnblogs.com
www.qq.com
www.douban.com
```

使用包含3个进程的进程池下载文件中网站的首页。(你可以使用subprocess调用wget或者curl等下载工具执行具体的下载任务)

共享资源

我们在Python多进程初步已经提到，我们应该尽量避免多进程共享资源。多进程共享资源必然会带来进程间相互竞争。而这种竞争又会造成race condition，我们的结果有可能被竞争的不确定性所影响。但如果需要，我们依然可以通过共享内存和Manager对象这么做。



共享“资源”

共享内存

在Linux进程间通信中，我们已经讲述了共享内存(shared memory)的原理，这里给出用Python实现的例子：



```
# modified from official documentation
import multiprocessing


def f(n, a):
```

```
n.value    = 3.14
a[0]       = 5

num    = multiprocessing.Value('d', 0.0)
arr    = multiprocessing.Array('i', range(10))

p = multiprocessing.Process(target=f, args=(num, arr))
p.start()
p.join()


print num.value
print arr[:]
```



这里我们实际上只有主进程和Process对象代表的进程。我们在主进程的内存空间中创建共享的内存，也就是Value和Array两个对象。对象Value被设置成为双精度数(d)，并初始化为0.0。而Array则类似于C中的数组，有固定的类型(i，也就是整数)。在Process进程中，我们修改了Value和Array对象。回到主程序，打印出结果，主程序也看到了两个对象的改变，说明资源确实在两个进程之间共享。

Manager

Manager对象类似于服务器与客户之间的通信 (server-client)，与我们在Internet上的活动很类似。我们用一个进程作为服务器，建立Manager来真正存放资源。其它的进程可以通过参数传递或者根据地址来访问Manager，建立连接后，操作服务器上的资源。在防火墙允许的情况下，我们完全可以将Manager运用于多计算机，从而模仿了一个真实的网络情境。下面的例子中，我们对Manager的使用类似于shared memory，但可以共享更丰富的对象类型。



```
import multiprocessing

def f(x, arr, l):
    x.value = 3.14
    arr[0] = 5
    l.append('Hello')

server = multiprocessing.Manager()
```

```
x      = server.Value('d', 0.0)
arr    = server.Array('i', range(10))
l      = server.list()

proc = multiprocessing.Process(target=f, args=(x, arr, l))
proc.start()
proc.join()

print(x.value)
print(arr)
print(l)
```



Manager利用`list()`方法提供了表的共享方式。实际上你可以利用`dict()`来共享词典，`Lock()`来共享`threading.Lock`(注意，我们共享的是`threading.Lock`，而不是进程的`mutiprocessing.Lock`。后者本身已经实现了进程共享)等。这样Manager就允许我们共享更多样的对象。

我们在这里不深入讲解Manager在远程情况下的应用。有机会的话，会在网络应用中进一步探索。

总结

Pool

Shared memory, Manager

标签: [Python](#)

好文要顶

关注我

收藏该文



Vamei

关注 - 26

粉丝 - 4985

荣誉: [推荐博客](#)

[+加关注](#)

5

0

(请您对文章做出评价)

« 上一篇: [Python标准库10 多进程初步 \(multiprocessing包\)](#)

» 下一篇: [谁动了我的奶酪?](#)

posted @ 2012-10-13 12:04 Vamei 阅读(19061) 评论(6) 编辑 收藏

评论列表

#1楼 2014-05-20 01:18 逍遥22

楼主,我在使用进程池的时候:代码如下:

```
1  def main():
2      host = '127.0.0.1'
3      port = 9999
4      servfd = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
5      servfd.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
6      servfd.bind((host, port))
7      servfd.listen(10)
8      while True:
9          try:
10             conn, addr = servfd.accept()
11             print conn, addr, type(conn)
12             process_Pool.apply_async(Proxy, (conn, addr,))
13         except:
14             pass
15
16
17  if __name__ == '__main__':
18      main()
19
20
```

其中Proxy为一个在当前py文件中的全局函数,但运行后出现这种错误:

```
1  Exception in thread Thread-2:
2  Traceback (most recent call last):
3      File "/usr/lib/python2.7/threading.py", line 808, in __bootstrap_inner
4          self.run()
5      File "/usr/lib/python2.7/threading.py", line 761, in run
6          self.__target(*self.__args, **self.__kwargs)
7      File "/usr/lib/python2.7/multiprocessing/pool.py", line 342, in _handle_tasks
8          put(task)
9  TypeError: expected string or Unicode object, NoneType found
```

这是为什么呢?思索了很久,很想知道为什么?

支持(0) 反对(0)

#2楼 2014-06-05 16:12 juji815

你好，楼主，我试验了文章中第一个例子，但是报错了，这是为什么呢？

Traceback (most recent call last):

File "<string>", line 1, in <module>

File "C:\Python33\lib\multiprocessing\forking.py", line 350, in main
prepare(preparation_data)

File "C:\Python33\lib\multiprocessing\forking.py", line 457, in prepare
'__parents_main__', file, path_name, etc

File "C:\Python33\lib\imp.py", line 175, in load_module
return load_source(name, filename, file)

File "C:\Python33\lib\imp.py", line 114, in load_source
_LoadSourceCompatibility(name, pathname, file).load_module(name)

File "<frozen importlib._bootstrap>", line 586, in _check_name_wrapper

File "<frozen importlib._bootstrap>", line 1024, in load_module

File "<frozen importlib._bootstrap>", line 1005, in load_module

File "<frozen importlib._bootstrap>", line 562, in
module_for_loader_wrapper

File "<frozen importlib._bootstrap>", line 870, in _load_module

File "<frozen importlib._bootstrap>", line 313, in
_call_with_frames_removed

File "E:\workspace\firstproject\1\2.py", line 6, in <module>
pool = mul.Pool(5)

File "C:\Python33\lib\multiprocessing__init__.py", line 222, in Pool
return Pool(processes, initializer, initargs, maxtasksperchild)

File "C:\Python33\lib\multiprocessing\pool.py", line 137, in __init__
self._repopulate_pool()

File "C:\Python33\lib\multiprocessing\pool.py", line 200, in
_repopulate_pool

w.start()

File "C:\Python33\lib\multiprocessing\process.py", line 111, in start
self._popen = Popen(self)

File "C:\Python33\lib\multiprocessing\forking.py", line 216, in __init__
cmd = ' '.join("%s" % x for x in get_command_line())

File "C:\Python33\lib\multiprocessing\forking.py", line 328, in
get_command_line

is not going to be frozen to produce a Windows executable."")

RuntimeError:

Attempt to start a new process before the current process has finished its bootstrapping phase.

This probably means that you are on Windows and you have forgotten to use the proper idiom in the main module:

```
if __name__ == '__main__':
    freeze_support()
...
```

The "freeze_support()" line can be omitted if the program is not going to be frozen to produce a Windows executable.

支持(0) 反对(0)

#3楼 2014-07-10 10:57 木冬淡雪

```
1  import multiprocessing as mul
2  import subprocess
3
4  def func(target):
5      rc = subprocess.call(["wget",target])
6
7  urls=[]
8  for line in open('urls.txt'):
9      urls.append(line.strip())
10
11 print urls
10 pool = mul.Pool(3)
11 pool.map(func,urls)
```

支持(3) 反对(0)

#4楼 2014-08-13 16:03 heroFuture

初学multiprocessing,文章很有用,谢谢楼主!

支持(0) 反对(0)

#5楼 2015-05-02 09:34 yuanyeh

```
1  #!/usr/bin/env python
2
```



```
3 import subprocess as sub
4 import multiprocessing as mul
5
6 # define the download function
7 def download(url):
8     url_name = url.split('.', 5)
9     url_name = url_name[1] + ".html"
10    cmd = "wget " + url + " -O " + url_name
11    sub.call(cmd, shell = True)
12
13 # read the url in the file
14 f = open("download.txt", "r")
15 lines = f.readlines();
16 f.close()
17
18 urls = []
19 for line in lines:
20     urls.append(line.strip())
21
22 # create six process
23 pool = mul.Pool(6)
24 pool.map(download, urls)
```

支持(0) 反对(0)

#6楼 2016-03-18 15:14 一小步

```
1 #!/usr/bin/env python
2 #coding=utf8
3
4 import os
5 import multiprocessing as mul
6
7 def f(x):
8     return str(x**2)+' pid:'+str(os.getpid())
9
10 pool = mul.Pool(5)
11 rel = pool.map(f,[1,2,3,4,5,6,7,8,9,10])
12 for id in rel:
13     print (id)
```

我做了下测试结果输出如下:

1 pid:5932

4 pid:9440
9 pid:9872
16 pid:1240
25 pid:5932
36 pid:9440
49 pid:9872
64 pid:1240
81 pid:5932
100 pid:9440

Pool(5)不是应该创建5个进程的吗，为什么打印出来的只有4个，是程序本身就占了一个吗。

刚刚开始学Python还是个菜鸟，求大神解答

支持(0) 反对(0)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问](#)网站首页。

【推荐】50万行VC++源码：大型组态工控、电力仿真CAD与GIS源码库

【推荐】融云即时通讯云—豆果美食、Faceu等亿级APP都在用



JPush 极光推送

消息推送领导品牌全面升级

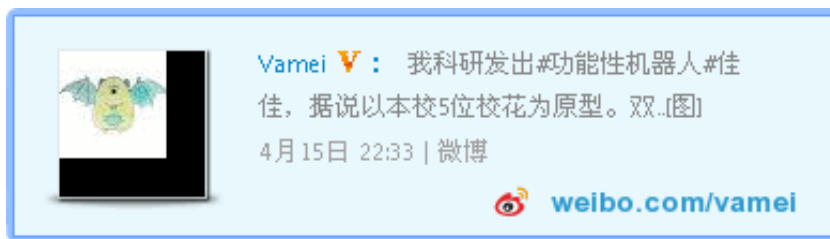


JIGUANG | 极光

[详情点击](#)

公告

你好，这里是Vamei，一名编程爱好者。我在博客里写了**Python/Linux/网络协议/算法/Java/数据科学**系列文章，[从这里开始阅读](#)。非常期待和你的交流。



我的微博

下列教程已经做成电子出版物，内容经过修订，也方便离线阅读：

协议森林

欢迎阅读我写的其他书籍：

现代小城的考古学家

天气与历史的相爱相杀

随手拍光影

昵称：Vamei

园龄：4年1个月

荣誉：推荐博客

粉丝：4985

关注：26

+加关注

常用链接

我的随笔

我的评论

我的参与

最新评论

我的标签

我的标签

Python(61)

Java(42)

大数据(22)

Linux(17)

网络(16)

算法(15)

文青(14)

技普(9)

系列索引(6)

开发工具(4)

更多

系列文章

Java快速教程

Linux的概念与体系

Python快速教程

数据科学

协议森林

纸上谈兵：算法与数据结构

积分与排名

积分 - 659668

排名 - 122

最新评论

1. Re:Java基础11 对象引用

受教！

--MissLost

2. Re:Python快速教程

看评论区一片喝彩！看来我得在此扎营了！

--测试小蚂蚁

3. Re:Python进阶06 循环对象

好好地列表解析变成了表推导

--ashic

4. Re:“不给力啊，老湿！”：RSA加密与破解

感谢楼主精彩分享

--worldball

5. Re:概率论04 随机变量

你写的这一系列太棒了，刚加入博客园就在你这里学到了，我要转载过去学习一下

--yixius

6. Re:Python基础03 序列

挺好的教程、、、

--王小拽的号

7. Re:Python进阶07 函数对象

```
def func(x,y): print x**ydef test(f,a,b): print 'test' print f(a,b)test (func,3,2)
```

输出的内容:tes.....

--M-edea

8. Re:Python进阶02 文本文件的输入输出

@coderXT换行符: \n...

--行者之印

9. Re:数据科学

博主啊，这里是一枚即将大二的计算机新人，大一学了python，java，还有一些算法，数据结构，图论了，感觉我对数学又一些反感，但是听说离散数学对计算机专业的很重要，不知道怎么去学比较好呢，我想像您写.....

--Acokil

10. Re:为什么要写技术博

楼主是用自己自定义的模板吗？在博客园里找不到这种风格的blog模板？

--行者之印

11. Re:来玩Play框架01 简介

挖煤哥,我补充了一下Windows下的搭建play框架,希望有点帮助,谢谢!

--Sungeek

12. Re:来玩Play框架07 静态文件

@helper.form(action = routes.Application.upload, 'enctype ->
"multipart/form-data") {--action = rout.....

--quxiaozha

13. Re:来玩Play框架07 静态文件

该记录将/assets/下的URL，对应到项目的/public文件夹内的文件。比如在项目的/public/images/test.jpg，就可以通过/assests/images/test.jpg这一.....

--quxiaozha

14. Re:来玩Play框架06 用户验证

支持挖煤哥~~~

--quxiaozha

15. Re:“不给力啊，老湿！”：RSA加密与破解

@maanshancss请你仔细阅读了这个文章再来评价。...

--Vamei

推荐排行榜

1. “不给力啊，老湿！”：RSA加密与破解(218)

2. Python快速教程(140)

3. 野蛮生长又五年(91)

4. Java快速教程(88)
5. 协议森林01 邮差与邮局 (网络协议概观)(79)
6. 为什么要写技术博(71)
7. 编程异闻录(54)
8. 博客一年：心理之旅(49)
9. 协议森林08 不放弃 (TCP协议与流通信)(45)
10. Python快速教程 尾声(43)
11. 协议森林(42)
12. Java基础01 从HelloWorld到面向对象(42)
13. Python基础08 面向对象的基本概念(40)
14. 一天能学会的计算机技术(34)
15. 博客第二年，杂谈(33)

Copyright ©2016 Vamei

05368581