Phantomjs性能优化

nmask / 2017-03-01 12:46:00 / 浏览数 4225 安全技术 技术讨论 顶(0) 踩(0)

写过爬虫的朋友应该都用过一个无头浏览器–phantomjs，使用它的原因很简单明了：能够高度模拟浏览器访问（对抗反爬虫），无头浏览（可以节省性能）。Phantomjs应

phantomjs也是爬虫界的一大神器，我最初使用它就是用来爬取一些动态加载的网页，效果俱佳。当然Phantomjs也不是完美无缺的，虽然作为无头浏览器其性能已经比其

关于phantomjs的安装使用网上一大推，这里也不在重复介绍，本篇文章重点在于介绍Phantomjs性能优化问题。因为我比较熟悉python语言，因此就借助此语言谈谈Phan

基础篇（设置参数功法）

Python中使用Phantomjs需要借助Selenium模块，Selenium本身也是用来做Web自动化测试的，正好封装了Phantomjs，因此我们可以借助它来使用Phantomjs。具体安

代码测试

访问单个网站的速度

默认配置：

```
from selenium import webdriver

d=webdriver.PhantomJS("D:\python27\Scripts\phantomjs.exe",service_args=[])
d.get("http://thief.one")
d.quit()
```

测试结果：3.2s

修改配置：

```
from selenium import webdriver

service_args=[]
service_args.append('--load-images=no')  ##■■■■■■■
service_args.append('--disk-cache=yes')  ##■■■■■
service_args.append('--ignore-ssl-errors=true') ##■■https■■

d=webdriver.PhantomJS("D:\python27\Scripts\phantomjs.exe",service_args=service_args)
d.get("http://thief.one")
d.quit()
```

测试结果：2.9s

说明：从单个网站来看，合理设置参数可以提速0.3s（如果网站上图片等资源较多，则提升的效果会更明显）。

设置超时

当利用爬虫访问一批网站时，遇到加载慢的网站往往会阻塞很久，遇到打不开的网站则会一直阻塞，严重影响了爬虫的性能，我们知道一般的爬虫，例如requests、urllib等

```
from selenium import webdriver

service_args=[]
service_args.append('--load-images=no')
service_args.append('--disk-cache=yes')
service_args.append('--ignore-ssl-errors=true')

d=webdriver.PhantomJS("D:\python27\Scripts\phantomjs.exe",service_args=service_args)
d.implicitly_wait(10)        ##■■■■■■■
d.set_page_load_timeout(10)  ##■■■■■■■
d.get("http://www.baidu.com")
d.quit()
```

说明：如果phantomjs加载时间超过10s，则会触发异常。（虽然触发异常，但current_url仍然可以用来获取当前url，源码也可以获取，只不过是没有加载完全的源码。当然

中级篇（合理开关）

在我使用phantomjs的一段时间内，通过不断调试，我发现phantomjs主要的性能消耗在于phantomjs进程的开启上。因为在python中使用phantomjs，相当于开启并调用

代码测试

单线程访问百度10次：

优化前：

```
from selenium import webdriver

def phantomjs_req(url):
   service_args=[]
   service_args.append('--load-images=no')
   service_args.append('--disk-cache=yes')
   service_args.append('--ignore-ssl-errors=true')

   d=webdriver.PhantomJS("D:\python27\Scripts\phantomjs.exe",service_args=service_args)
   d.get(url)
   print d.current_url

   d.quit()

url_list=["http://www.baidu.com"]*10
for i in url_list:
   phantomjs_req(i)
```

测试结果：28.2s，运行过程中,phantomjs进程不断开关。

优化后：

```
from selenium import webdriver

def phantomjs_req(url):
   d.get(url)
   print d.current_url

service_args=[]
service_args.append('--load-images=no')
service_args.append('--disk-cache=yes')
service_args.append('--ignore-ssl-errors=true')
d=webdriver.PhantomJS("D:\python27\Scripts\phantomjs.exe",service_args=service_args)

url_list=["http://www.baidu.com"]*10
for i in url_list:
   phantomjs_req(i)

d.quit()
```

测试结果：4.2s

说明：可以看到优化前与优化后代码的区别，在于将phantomjs开启关闭的操作放到了循环外面，使它始终只开关一次。可以看到性能的差别非常大，因此也可以看出phan

注意：此方法虽然节省了很大的开支，但会引起另外一个phantomjs的Bug（暂且称之为Bug），也就是phantomjs状态覆盖问题。当批量去访问一些网站时，会发现返回的

高级篇（phantomjs并发问题）

通过前面的优化，我们发现phantomjs的性能提高了很多，但即便如此，以上代码也只是实现了单线程中的优化。当遇到大批量的网站时，并发是必须的选择，那么Phanto

优化之路

在优化phantomjs并发性能的问题上，我也并没有一帆风顺，期间查阅了很多资料，也踩过了很多的坑。

不成熟的优化（一）

起初我用了最直接了当的方法，企图开启phantomjs并发的性能。（运行一个phantomjs进程，进程内开启多线程）

```
d=webdriver.PhantomJS()
def test(url):
    d.get(url)

url_list=["http://www.baidu.com"]*10
for url in url_list:
    threading.Thread(target=test,args=(url,)).start()
d.quit()
```

然而运行连连出错，在查看了官网等资料后发现phantomjs是单线程的，因此如果按照上面的写法，那么不能使用多线程同时去执行，此次优化失败！

不成熟的优化（二）

既然一个phantomjs只能支持单线程，那么我就开启多个phantomjs。

```
def test(url):
    d=webdriver.PhantomJS()
    d.get(url)
    d.quit()

url_list=["http://www.baidu.com"]*10
for url in url_list:
   threading.Thread(target=test,args=(url,)).start()
```

终于我看到同时10个phantomjs进程被开启了，10个网站的请求可以并发执行了。然而当网站的数量为50个时，要同时运行50个phantomjs进程？No，这必定会搞垮服务

不成熟的优化（三）

经过以上2次失败，我开始思考，如何只开启10个phantomjs进程，然后每个phantomjs进程按顺序执行请求网站的操作呢?这样就相当于10个进程并发执行了。
终于在某个夜晚，我想出了以下代码：

```
def  test():
    d=webdriver.PhantomJS()
    for i in url_list:
       d.get(url)
    d.quit()

url_list=["http://www.baidu.com"]*50
for i in range(10):
    threading.Thread(target=test).start()
```

成功开启了10个phantomjs进程，每个进程按顺序执行了50个网站的请求。等等，貌似这样设计，每个phantomjs进程都会去访问50次百度，这不是最初的要求，oh，No

不算成熟但还可以的优化

在第三阶段并发优化的雏形已经出来了，只不过还需要解决一个多线程共享资源的问题，这个可以用Queue模块解决。那么直接看优化后并发的代码：

```
__author__="nMask"
__Date__="20170224"
__Blog__="http://thief.one"

import Queue
from selenium import webdriver
import threading
import time

class conphantomjs:
   phantomjs_max=1             ##■■■■phantomjs■■
   jiange=0.00001                  ##■■phantomjs■■
   timeout=20                  ##■■phantomjs■■■■
   path="D:\python27\Scripts\phantomjs.exe" ##phantomjs■■
   service_args=['--load-images=no','--disk-cache=yes'] ##■■■■

   def __init__(self):
       self.q_phantomjs=Queue.Queue()   ##■■phantomjs■■■■

   def getbody(self,url):
       '''
       ■■phantomjs■■■■■■■■url
       '''
       d=self.q_phantomjs.get()

       try:
           d.get(url)
       except:
           print "Phantomjs Open url Error"

       url=d.current_url

       self.q_phantomjs.put(d)
```

```
        print url

    def open_phantomjs(self):
        '''
        ■■■■■phantomjs■■
        '''
        def open_threading():
            d=webdriver.PhantomJS(conphantomjs.path,service_args=conphantomjs.service_args)
            d.implicitly_wait(conphantomjs.timeout)        ##■■■■■■
            d.set_page_load_timeout(conphantomjs.timeout)  ##■■■■■■

            self.q_phantomjs.put(d) #■phantomjs■■■■■

        th=[]
        for i in range(conphantomjs.phantomjs_max):
            t=threading.Thread(target=open_threading)
            th.append(t)
        for i in th:
            i.start()
            time.sleep(conphantomjs.jiange) #■■■■■■■■■
        for i in th:
            i.join()


    def close_phantomjs(self):
        '''
        ■■■■■phantomjs■■
        '''
        th=[]
        def close_threading():
            d=self.q_phantomjs.get()
            d.quit()

        for i in range(self.q_phantomjs.qsize()):
            t=threading.Thread(target=close_threading)
            th.append(t)
        for i in th:
            i.start()
        for i in th:
            i.join()


if __name__=="__main__":
    '''
    ■■■
    1.■■■■
    2.■■open_phantomjs ■■phantomjs■■
    3.■■getbody■■■■■url
    4.■■close_phantomjs ■■phantomjs■■
    '''
    cur=conphantomjs()
    conphantomjs.phantomjs_max=10
    cur.open_phantomjs()
    print "phantomjs num is ",cur.q_phantomjs.qsize()

    url_list=["http://www.baidu.com"]*50

    th=[]
    for i in url_list:
        t=threading.Thread(target=cur.getbody,args=(i,))
        th.append(t)
    for i in th:
        i.start()
    for i in th:
        i.join()

    cur.close_phantomjs()
    print "phantomjs num is ",cur.q_phantomjs.qsize()
```

代码测试：

利用单线程优化后的代码访问50次百度：10.3s。
利用10个phantomjs并发访问50次百度：8.1s

说明：并发优化后的代码同时开启了10个phantomjs进程，用于处理50次访问百度的请求。由于一个phantomjs同一时间不能处理2个url，也就是说不支持多线程处理，因

终极篇

高级篇中解决并发效率，我用的实际上是多进程，无论python同时开启多少个线程去让phantomjs进程执行操作，一个phantomjs进程同时也只能执行一个访问请求。因此
既然知道了性能的瓶颈所在，那么终极篇中，我们可以使用分布式+phantomjs多进程并发来提高性能。

替代方案

以上的优化方案并不能从根本上解决phantomjs性能问题，更好的替代方案请移步：
Phantomjs正确打开方式

上一篇：Exec OS Command V... 下一篇：Mysql数据库反弹端口连接提权
1. 9 条回复

 admin 2017-03-02 06:43:16

用popen代替Selenium会不会提速呢

0 回复Ta

 hades 2017-03-02 07:37:20

博主最近python上瘾 哈哈 社区有位ADO大神可以一起聊聊

0 回复Ta


cover 2017-03-02 09:01:36

兄弟看你对phantomjs踩的坑还是少了，这些算不上太大的问题，等你遇到浏览器的能打开的网站，phantomjs打不开的网站，你就懵逼了

0 回复Ta


nmask 2017-03-03 01:02:08

没办法，最近公司项目要用到python开发，只能自己瞎捉摸了。

0 回复Ta


nmask 2017-03-03 01:03:02

嗯嗯，phantomjs用得还不算多，还要继续爬坑啊。phantomjs问题有没有一个比较好的学习资料呢？感觉这方面的坑还是很多的。

0 回复Ta


nmask 2017-03-03 01:03:44

倒没试过用popen去启动

0 回复Ta


cover 2017-03-03 01:30:11

不建议用phantomjs做爬虫，我们做扫描器就用phantomjs做的爬虫， 经常爆一些time out的错误，还有一些看不懂的错误，过段时间直接重写，这个包不靠谱

0 回复Ta


nmask 2017-03-03 08:30:34

我也是一直碰到timeout。。。。。。

但是能模拟浏览器，且又是无头的还有其他的选择吗？

请教下，还有其他什么包可以用呢？

0 回复Ta


jiajiaozhon****@ 2019-05-27 17:08:37

这个轮子，现在是不是不能跑了？url 没有打开。

0 回复Ta

登录 后跟帖

先知社区

现在登录

热门节点

技术文章

社区小黑板

目录

RSS 关于社区 友情链接 社区小黑板