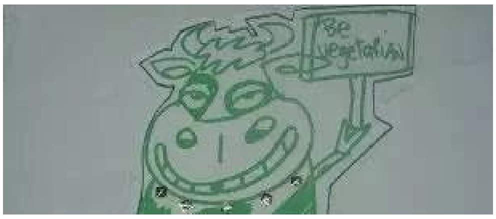

curl_cffi开启gevent协程

原创 邓世龙 静夜随想

平时使用requests和gevent写爬虫的时候，是如下demo, 在代码最前面加上monkey.patch_all()即可

```python
import gevent
from gevent import pool, monkey
monkey.patch_all()
import urllib3
urllib3.disable_warnings(urllib3.exceptions.InsecureRequestWarning)
import time
import requests
from gevent.lock import RLock

key_lock = RLock()
key = 0


def get_key():
    global key
    key_lock.acquire()
    key += 1
    key_lock.release()
    return key

def test_request():
    print("start ")
    time.sleep(2)
    print(get_key())
    session = requests.session()
    response = session.get('https://www.baidu.com', verify=False, timeout=10)
    print(response.status_code, response.url)
    time.sleep(3)
    print('end')



if __name__ == "__main__":
    n = 3
    gevent_poll = pool.Pool(n)
    jobs = []
    for i in range(n):
        job = gevent_poll.spawn(test_request)
        jobs.append(job)
    gevent.joinall(jobs)
```

输出结果里 0 1 2 是连续的, 请求是并发的

```
0
1
2
200 https://www.baidu.com/
200 https://www.baidu.com/
200 https://www.baidu.com/
```

为了过ja3指纹，我们可以使用curl_cffi这个库，demo如下

```python
import gevent
from gevent import pool, monkey
monkey.patch_all()
import urllib3
urllib3.disable_warnings(urllib3.exceptions.InsecureRequestWarning)
import time
import requests
from curl_cffi import requests as cffi_requests
from gevent.lock import RLock

key_lock = RLock()
key = 0


def get_key():
    global key
    key_lock.acquire()
    key += 1
    key_lock.release()
    return key

def test_request():
    print("start ")
    time.sleep(2)
    print(get_key())
    session = cffi_requests.Session()
    response = session.get('https://www.baidu.com', verify=False, timeout=10)
    print(response.status_code, response.url)
    time.sleep(3)
    print('end')


if __name__ == "__main__":
    n = 3
    gevent_poll = pool.Pool(n)
    jobs = []
    for i in range(n):
        job = gevent_poll.spawn(test_request)
        jobs.append(job)
    gevent.joinall(jobs)
```

但这样并不能实现并发，输出结果里0 1 2是分开的，请求是顺序的

```
0
200 https://www.baidu.com/
1
200 https://www.baidu.com/
2
200 https://www.baidu.com/
```

解决办法是在创建session时加上thread参数，指定gevent即可，像session = cffi_requests.Session(thread='gevent')这样创建即可，最终代码如下

```
from gevent import pool, monkey
monkey.patch_all()
import urllib3
urllib3.disable_warnings(urllib3.exceptions.InsecureRequestWarning)
import time
import requests
from curl_cffi import requests as cffi_requests
from gevent.lock import RLock

key_lock = RLock()
key = 0


def get_key():
    global key
    key_lock.acquire()
    key += 1
    key_lock.release()
    return key

def test_request():
    print("start ")
    time.sleep(2)
    print(get_key())
    session = cffi_requests.Session(thread='gevent')
    response = session.get('https://www.baidu.com', verify=False, timeout=10)
    print(response.status_code, response.url)
    time.sleep(3)
    print('end')


if __name__ == "__main__":
    n = 3
    gevent_poll = pool.Pool(n)
    jobs = []
    for i in range(n):
        job = gevent_poll.spawn(test_request)
        jobs.append(job)
    gevent.joinall(jobs)
```

结果里0 1 2 是顺序的，请求是并发的。

```
0
1
2
200 https://www.baidu.com/
200 https://www.baidu.com/
200 https://www.baidu.com/
```

这又是一个小细节，被坑了。正常以为会像requests库一样用，但结果不是。文档里有说到这个用法，但没仔细去看了，这个库API和requests太像了，以为可以无缝衔接。

参考资料

https://curl-cffi.readthedocs.io/en/latest/advanced.html#using-with-eventlet-gevent