



PyCon 2014

China

北京 上海 珠海 苏州 杭州

主办: PyChina.org, GDG

万万没想到之： Python Web异步编程记

jeff@toraysoft.com

本栏目由“珠三角技术沙龙”赞助播出



蕉爷，你姓蕉吗？ Yeah! 爷姓蕉!

本栏目由“音乐说”赞助播出



音乐说

有声朋友圈 遇见陌声人

觉得我好听就闪聊一下呗

本栏目由“萌叔漫玩”赞助播出



小妹妹，你看大叔萌萌哒，有木有

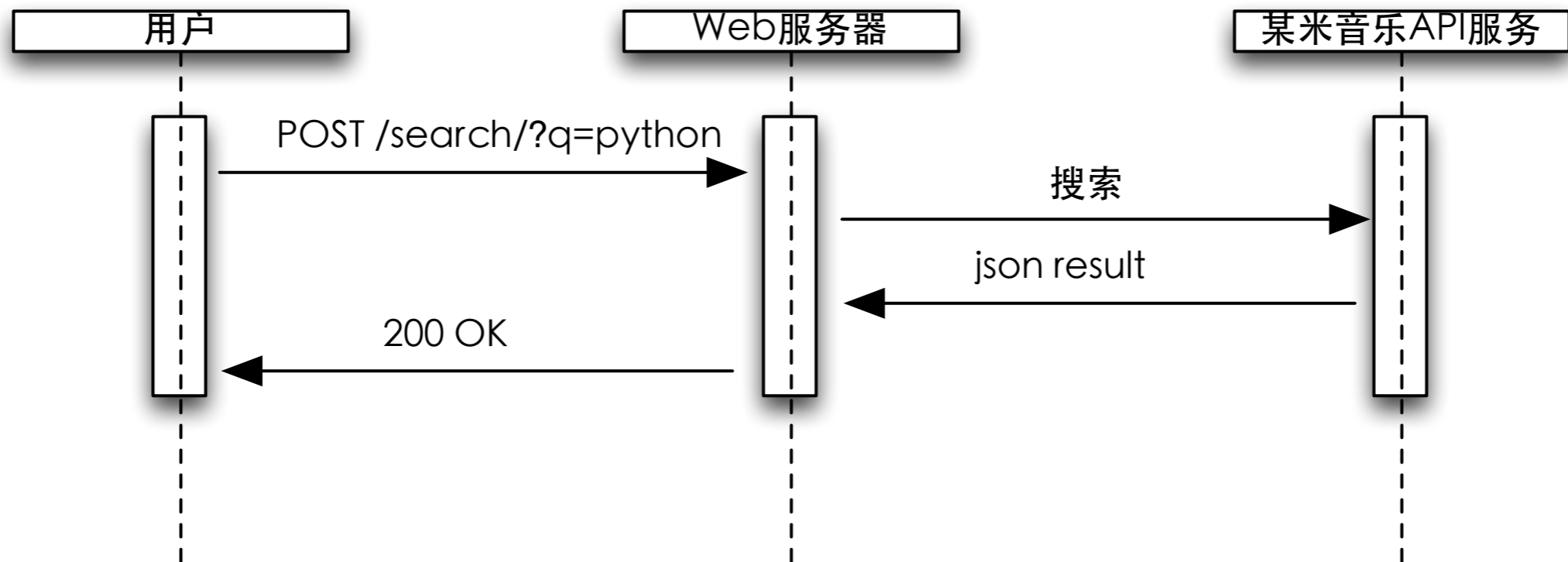
@jeff_kit
@姐夫

前珠三角技术沙龙首席组委
Toraysoft co-founder
Toraysoft 首席码农



举个栗子开始

听题：写一个搜索歌曲的接口



Django 一下

```
def search(request):  
    q = request.GET.get('q', '')  
    rsp = requests.get('http://baidu.com/s?word=' + q)  
    return HttpResponse(rsp.text)
```

```
uwsgi --http :8000 --module pyconf2014.wsgi --processes 1
```

```
9:34上午 0:00.00 uwsgi --http :8000 --module pyconf2014.wsgi --home /Users/jeff/.virtualenvs/
```

```
9:34上午 0:00.20 uwsgi --http :8000 --module pyconf2014.wsgi --home /Users/jeff/.virtualenvs/
```


ab之

```
jeff@jeffkit-2~/projects/music/music-web$ab -c 100 -n 100 http://127.0.0.1:8000/search/
This is ApacheBench, Version 2.3 <$Revision: 1554214 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 127.0.0.1 (be patient).....done

Server Software:
Server Hostname:      127.0.0.1
Server Port:          8000

Document Path:        /search/
Document Length:      8 bytes

Concurrency Level:    100
Time taken for tests: 59.367 seconds
Complete requests:    100
Failed requests:      89
  (Connect: 0, Receive: 0, Length: 89, Exceptions: 0)
Total transferred:    1056 bytes
HTML transferred:     88 bytes
Requests per second:  1.68 [#/sec] (mean)
Time per request:     59366.775 [ms] (mean)
Time per request:     593.668 [ms] (mean, across all concurrent requests)
Transfer rate:         0.02 [Kbytes/sec] received
```

500秒!

再ab之

```
jeff@jeffkit-2~/projects/music/music-web$ab -c 100 -n 100 http://127.0.0.1:8000/search/
This is ApacheBench, Version 2.3 <$Revision: 1554214 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 127.0.0.1 (be patient).....done

Server Software:
Server Hostname:      127.0.0.1
Server Port:         8000

Document Path:       /search/
Document Length:     8 bytes

Concurrency Level:   100
Time taken for tests: 50.824 seconds
Complete requests:   100
Failed requests:     0
Total transferred:   9600 bytes
HTML transferred:    800 bytes
Requests per second: 1.97 [#/sec] (mean)
Time per request:    50823.551 [ms] (mean)
Time per request:    508.236 [ms] (mean, across all concurrent requests)
Transfer rate:       0.18 [Kbytes/sec] received
```

50秒!

嗯！启动100条进程就
可以了！

问题在哪？

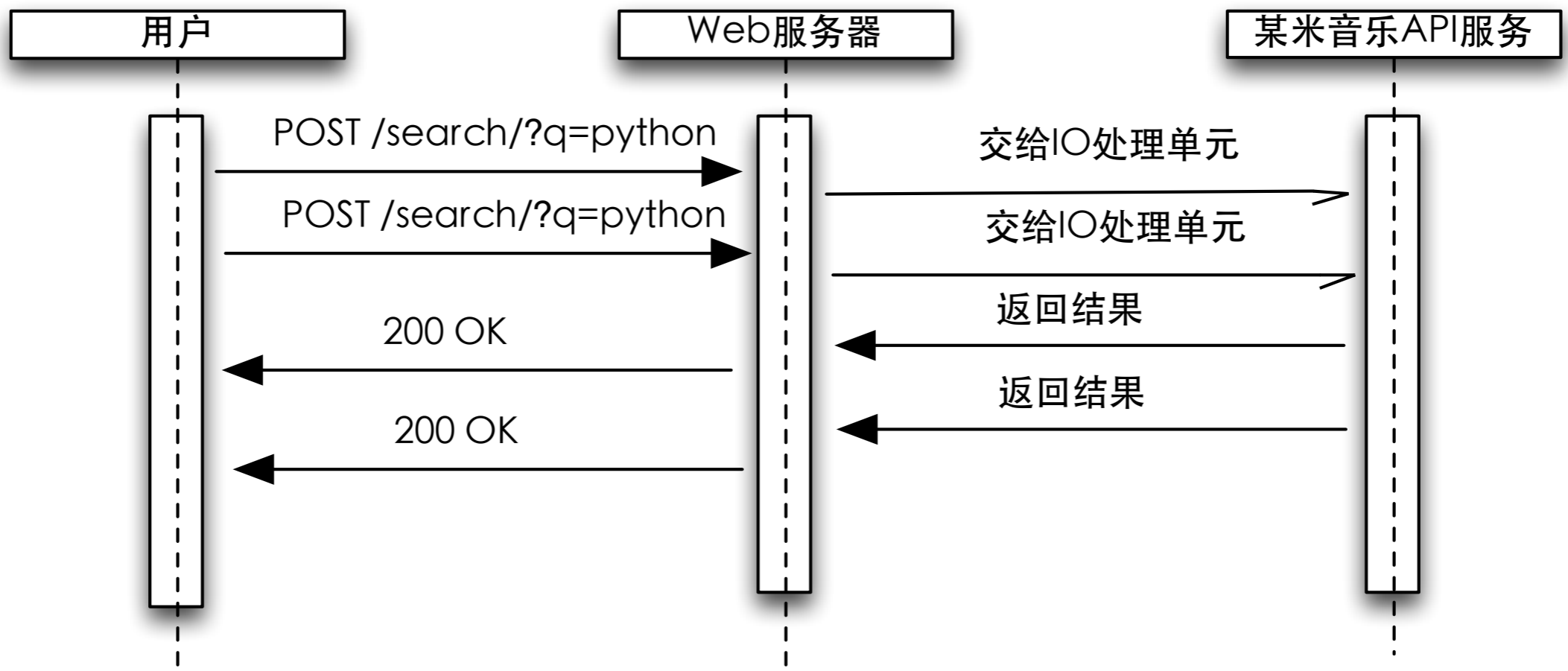
- 同步网络请求耗时长，前面的请求导致后续的请求阻塞。
- 最大进程数就是最大并发数。
- 消耗巨大的服务器资源。
- 请求队列爆掉，如同被DDOS！

异步解决方案

- 非阻塞IO + 异步调度
- select, epoll, kqueue

说人话

- web服务器请求第三方服务的时候，交出控制权，返回去受理新的用户请求。
- 第三方服务有返回的时候，通知web服务器进程，把结果返回给用户。



异步web框架哪家强？

- 没有异步web框架，wsgi是同步标准。
- 那是web server的事儿。

异步webserver哪家强？

- twisted，超重量级。略。
- Tornado，既是server，也是framework。
- Gevent兼容的uwsgi Contailer， uWsgi， Gunicorn。

Tornado

```
class SearchHandler(tornado.web.RequestHandler):  
    @asynchronous  
    def get(self):  
        http_client = AsyncHTTPClient()  
        http_client.fetch("http://music.xxx.com/?q=python",  
                          callback=self.on_fetch)  
  
    def on_fetch(self, response):  
        self.write(response.body)  
        self.finish()
```

作为Server

```
application = tornado.web.Application([
    (r"/search", SearchHandler),
])

if __name__ == "__main__":
    application.listen(8888)
    tornado.ioloop.IOLoop.instance().start()
```

```
python run_tornado.py
```

```
4230 0.0 0.1 2470504 12000 s003 S+ 2:20下午 0:00.14 python run_tornado.py
```

ab一下

```
100% 126 (longest request)
jeff@jeffkit-2~/projects/music/music-web$ ab -c 100 -n 100 http://127.0.0.1:8888/search
This is ApacheBench, Version 2.3 <$Revision: 1554214 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 127.0.0.1 (be patient).....done

Server Software:      TornadoServer/4.0.2
Server Hostname:     127.0.0.1
Server Port:         8888

Document Path:       /search
Document Length:     2 bytes

Concurrency Level:   100
Time taken for tests: 5.149 seconds
Complete requests:  100
Failed requests:     0
Total transferred:  19600 bytes
HTML transferred:   200 bytes
Requests per second: 19.42 [#/sec] (mean)
Time per request:   5149.307 [ms] (mean)
Time per request:   51.493 [ms] (mean, across all concurrent requests)
Transfer rate:      3.72 [Kbytes/sec] received
```

5秒!

回调? 无爱! ~~耶!~~ yield!

```
class SearchHandler2(tornado.web.RequestHandler):  
  
    @gen.coroutine  
    def get(self):  
        http_client = AsyncHTTPClient()  
        response = yield http_client.fetch("http://music.xxx.com/?q=python")  
        self.write(response.body)  
        self.finish()
```

```
AsyncHTTPClient.configure(  
    'tornado.curl_httpclient.CurlAsyncHTTPClient',  
    max_clients=100)
```

优点

- 单线程，高并发
- 低能耗，高效率

可惜

- 回调模型无爱
- yield伪同步模式入侵性太强（第三方sdk要重新实现）
- server与框架耦合度太高，入侵性太强+1，不能使用熟悉的框架作业。
- 作为wsgi server，未能发挥其优势

Gevent!

- Greenlet
- libEvent
- monkey_patch()!

认识Gevent

```
import gevent
import socket
from gevent import monkey; monkey.patch_socket()

urls = ['www.google.com', 'www.facebook.com', 'www.python.org']
jobs = [gevent.spawn(socket.gethostbyname, url) for url in urls]
gevent.joinall(jobs, timeout=2)
[job.value for job in jobs]
##### out #####
['74.125.79.106', '208.77.188.166', '82.94.164.162']
```

Gevent能

- 让你按原来的方式写代码。
- 让同步编写的代码，异步地执行。
- 让单线程也并发，且比多线程更高效环保。
- 让原生的thread变成green thread。

Gevent + wsgi

- uWsgi



uWsgi

- 修改处理器的类型即可 processes => gevent
- patch all: — gevent-monkey-patch

代码，还是原来的味

```
def search(request):  
    q = request.GET.get('q', '')  
    rsp = requests.get('http://baidu.com/s?word=' + q)  
    return HttpResponse(rsp.text)
```

```
uwsgi --http :8000 --module pyconf2014.wsgi  
--gevent 100 --gevent-monkey-patch
```

再ab一下

```
100% 210 (longest request)
jeff@jeffkit-2~/projects/music/music-web$ab -c 100 -n 100 http://127.0.0.1:8000/search/
This is ApacheBench, Version 2.3 <$Revision: 1554214 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 127.0.0.1 (be patient).....done

Server Software:
Server Hostname:      127.0.0.1
Server Port:          8000

Document Path:        /search/
Document Length:      8 bytes

Concurrency Level:    100
Time taken for tests:  5.393 seconds
Complete requests:    100
Failed requests:      0
Total transferred:    9600 bytes
HTML transferred:     800 bytes
Requests per second:  18.54 [#/sec] (mean)
Time per request:     5393.336 [ms] (mean)
Time per request:     53.933 [ms] (mean, across all concurrent requests)
Transfer rate:        1.74 [Kbytes/sec] received
```

5秒!

但，uWsgi + Django的泪

- uWsgi的对thread的patch有问题
- 如果django的view有访问数据库，则数据库连接非线程（协程）安全。
- databasewrapper objects created in a thread can only be used in that same thread。

Gunicorn + Django

- Gunicorn, patch完美
- 但, 使用Django数据库连接池需谨慎! 高并发导致占用大量的数据库连接。

万万没想到

python web异步化，配个参数就完了

谢谢

