

Python是否适合高性能web开发

温铭

高性能服务端的要素

- 多：高并发，支撑用户多
- 快：单个请求处理速度快
- 易：易开发，易维护，方便调岗和招人

技术实现

- 高并发：异步
- 处理快：缓存，减少IO
- 易开发：库，只用关心逻辑

通用的架构

- Nginx -> py -> memcached(redis) -> mysql
- 同步
- 跨服务器请求

理想的架构

- 异步、同步语法
- 多级缓存：进程、本机kv、跨服务器kv、DB
- 没有网络IO

现实：异步

- *Tornado*: 第三方库齐全，回调
- *Gevent*: 不改代码
- *Asyncio* New in python 3.4

现实：缓存

- `functools.lru_cache` *New in version 3.2*

Python的优势

- 丰富的库
- 开发快
- 易上手

嫁接

- Python + ?

嫁接

- Openresty: 替换nginx, 提供异步和共享缓存
- Nginx -> py -> memcached(redis) -> mysql
- openresty-> py -> cache(openresty) -> kv(openresty) -> mysql(openresty)

自己写服务端？

- 用户注册、登录
- 短信通知
- 消息推动
- 聊天
- 存储
- 支付
- ...

不用写服务端

- Parse
- LeanCloud(AVOS)

Q&A



Ming 

广东 珠海



扫一扫上面的二维码图案，加我微信

