

云风的 BLOG

思绪来得快去得也快，偶尔会在这里停留

[« skynet 消息服务器支持](#) | [返回首页](#) | [设计一种简化的 protocol buffer 协议](#) »

给 skynet 增加 http 服务器模块

一直没给 skynet 加 http 协议解析模块是因为这个领域我不熟悉，而懂这块（web 开发）的人很多，随便找个人做都应该比我做的好。世界上的 web 服务器实在是太多了，足见做一个的门槛也不高，我也没什么需求，所以就这样等着有需要的人来补上这一块。

但这两天实在等不了了。我们即将上线的一款游戏，运营方要求我们提供一组 web api 供运营使用。固然我们可以单独写一个进程挂在 nginx 的后面，并和 skynet 通讯，但游戏开发组的同学觉得不必把简单的事情做的这么绕。监听一个端口提供 http 协议的服务又不是什么特别麻烦的事情，结果就打算直接在 skynet 里提供。

之前已经有非我们公司的同学在用 skynet 的时候自己写过一个 [http 服务器](#)。我觉得没有必要把 http 协议解析的部分和 socket 管理的部分搅到一起。我更希望有一个纯粹的 http 协议的应答模块。

在网上找不到合适的 lua 版的 http 协议解析模块，我便花了一个晚上，我实现了大概。

我希望外部只提供两个 api 注入：一个用于读一个数据流，另一个用于把数据写到数据流上。由于 skynet/lua 有很不错的 coroutine 支持，这组 api 便可以以阻塞模式提供，而不需要搞成 node.js 那样的 callback 也可以充分利用 cpu。

在 dev 分支上，[有这个 httpd 的实现](#)，也就是 200 来行代码吧。

对 skynet socket 的封装[放在这里](#)，它仅仅是在 socket 的 read 和 write 的基础上，把错误返回变成了抛出异常。

依靠这些，我搭建了一个简单的 web server，[见 skynet 的 examples](#)。这个范例监听了一个端口，并启动了 20 个 agent(worker) 等待 http 请求，这些 agent 会轮流接管新到的请求，使用 httpd 库解吸协议并回应。

这个库仅考虑了最基本的安全问题：http 请求非法、http header 过大都会主动返回（而不会让恶意攻击者制造一个特别的数据流来空耗服务器内存或占用服务器的 CPU）。使用这个库的时候，也可以设置 http body 的上限（如果不需要接受 client POST 的大文件的话）来限制从客户端发送过长的请求。

我没有做细致的性能测试，因为这个 web 服务本来就不是面对最终用户的，不会有什么性能上的要求。但有同学反馈说，他们编写的服务，在他的开发机（一台普通的 4 核台式机）上可以有 4K+ qps 的处理能力，应该是满不错了。当然，我一直对 skynet 的性能还是颇有信心的。

云风 提交于 July 23, 2014 08:18 PM | [固定链接](#)

COMMENTS

可以参考python的wsgi标准的gevent实现。

Posted by: www | (4) [July 24, 2014 10:05 AM](#)

可以参考python的wsgi标准的gevent实现。

Posted by: www | (3) [July 24, 2014 10:02 AM](#)

楼上掉书袋有什么意思，自己看一眼200行代码不就清楚了？

Posted by: [Dreampuf](#) | (2) [July 23, 2014 10:56 PM](#)

对持久连接、multi-range和chunked-transfer-encoding的支持如何？

Posted by: anonymous | (1) [July 23, 2014 08:59 PM](#)

POST A COMMENT

名字:

Email 地址:

为了验证您是人类, 请将六加一的结果 (阿拉伯数字七) 填写在下面:

URL:

☐ 记住我的信息?

留言:

(不欢迎在留言中粘贴程序代码)

提交