

# 云风的 BLOG

思绪来得快去得也快，偶尔会在这里停留

[« RLA 文件中的法线信息提取 | 返回首页 | 策划们离不开的 Excel »](#)

## skynet 的 UDP 支持

考虑了很久，最终还是给 skynet 加上了基本的 UDP 支持。虽然大多数情况下，我不赞成使用 UDP 协议。尤其是在网络游戏领域。但考虑到 skynet 已经不仅仅应用于游戏领域，我想，加入有限的 UDP 支持是有意义的。

btw, 根据最近的反馈看，有人把 skynet 用于交换机（由于使用的是 powerpc 的 CPU，帮助解决了一些大小端 bug）；有应用于证券领域；还有做视频广播的。另外，把 skynet 用于 web 开发的应该也有人在，就简单的测试来看，性能方面不比把 lua 集成到 nginx 差。

目前，UDP 这部分代码已经完成，放在 github 里一个叫 udp 的分支上，不久以后会合并到主干上。由于我自己没有什么这方面的需求，所以还需要有 udp 需求的同学读一下代码，实际使用，这样才可能发现潜在的问题。

关于这部分的设计以及 api 文档，[我补充在 wiki 上了](#)。

---

最近对网络消息分发做了点小优化，方法值得记录一下：

当有若干消息需要处理时，简单的方案是把消息放到一个队列里，然后依次从队列取出来处理。

但这么做有一个小问题：单条的消息处理过程可能被阻塞挂起，如果等待一条消息处理完再处理下一条就会因为上一条消息的阻塞而延缓之后的消息处理。

为了避免这种情况，我们可以用 skynet.fork 启动多个 coroutine 来分别处理队列消息。但这样做会增加许多 coroutine。如果消息处理流程并不会阻塞，这种方法又有点浪费（本可以用一个 coroutine 搞定）。

最好的方法是：一旦 coroutine 挂起，就立刻开启新的 coroutine 继续处理队列中的消息；如果没有挂起而是处理完了，就接着处理下一条。但是，这样就需要修改调度器了。

我想到一个折中的方案：

每次判断队列是否为空，如果不为空，就调用 skynet.fork 出自己这个函数处理这个消息队列。并接下来马上循环处理消息队列中的消息。

这样，如果消息处理中不会挂起的话，fork 出来的 coroutine 会发现队列已经为空立刻退出；如果消息处理挂起，那么 fork 的 coroutine 会接着处理没处理的消息。由于消息队列只有一个，同一条消息不会被 pop 两次，所以整个流程是正确的（且保证了消息处理开始时点的次序）。

创建的 coroutine 的个数也仅比上面提到的理想方案多一个而已。

---

最后，计划在今年内组织一次 skynet 的线下交流活动。地点在广州，时间未定。有兴趣的同学可以去[邮件列表](#)报名。

云风 提交于 November 13, 2014 04:27 PM | [固定链接](#)

## COMMENTS

队列这块能否详细说说，我之前也碰到过类似问题。

后来还是用了多个协程来处理了。

虽然浪费一些资源，但能应对突然极速增长的需求。

另外，你调度这块咋处理的，如果没有协程需要做的事情，等多久再查看一次？

---

Posted by: yeasy | (3) [December 1, 2014 11:21 AM](#)

请问斗罗大陆ol什么时候开测啊

---

Posted by: 李三 | (2) [November 23, 2014 09:17 PM](#)

用于证券领域...这也行啊...  
能否求下链接[好像不一定能拿到吧?]

Posted by: pzz2011 | (1) [November 16, 2014 08:34 PM](#)

## POST A COMMENT

非这个主题相关的留言请到: [留言本](#)

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