

# 云风的 BLOG

思绪来得快去得也快，偶尔会在这里停留

[« Hive , Lua 的 actor 模型 | 返回首页 | Hive 增加了 socket 库 »](#)

## skynet 下的用户登陆问题

今天收到一个朋友的邮件，他们使用 skynet 框架的游戏上线后遇到一些问题。

引用如下：

我有一个线上的 server 用了skynet框架，用了gate watchdog agent这样的结构，当人数到达1000左右的时候发现很多新 socket 连接都阻塞在了 gate 或 watchdog 这里，导致用户登录不了，后面发现是客户端那边在登录的时候做了超时判断，如果超时了就断开连接，重新连 server 执行登录过程，因为 watchdog 是在收到客户端发来的第一个消息的时候创建agent，这就导致很多 cpu 消耗在创建销毁 agent 上，而 agent 服务创建的代价比较高，这就导致watchdog效率下降。后来取消掉了客户端的那个登录超时判断，情况有所好转，但在人多的时候延迟还是存在，接着又优化了 server 这边的一些逻辑代码，现在还在观察中。

中午吃饭的时候，我和我们的开发人员讨论了这个问题。为什么我们在 1000 人级别时没有出现类似状况呢？我总结的原因如下：

我们把用户认证过程放在了 agent 启动之前，是由 watchdog 进行的。

这样，未认证连接在认证前是不会启动新的 agent 的。这避免了远超过系统承载能力的并发连接引起的问题。

这样做，会让 watchdog 的逻辑复杂一些，它需要先统一处理 gate 转发过来的数据包，直到认证结束，然后通知 gate 做包转发。watchdog 承载的任务也多一些，但这么做看起来是值得的。

agent 启动是一个 cpu 开销很大的操作。skynet 的 service 对象，本身是很轻量的。lua vm 的创建也不会有什么开销。但是在一个新的 lua vm 上初始化 lua 代码却会消耗大量的时间。

如果 agent 的启动全部由 watchdog 完成，就必然被串行化，并会阻塞住 watchdog 消化其它包。当多人同时登陆时势必是一个性能瓶颈。尤其是每个新建连接就立刻创建新的 agent 的策略，当用户登陆不上反复重试，更会将问题恶化。

最后，我修改了一下 skynet 中 lua 服务的启动流程，可能会对这种情况有所改善。

我将 lua 服务的启动流程改为两步，先创建出空的 lua vm 。然后注册一个专用于启动的消息处理函数，并立刻给自己发一个启动消息。这个消息一定是消息队列里的第一个消息。

接下来由这个启动消息来触发 lua vm 的进一步初始化过程。这样，就可以充分利用多核来处理并发登陆请求了。

这个修改可能引起的问题是，启动服务的调用无法侦测到启动失败。

最后，我认为，针对并发登陆的排队处理总是要做的。由 watchdog 转发登陆认证请求到一个排队服务器中去应该是一个更好的解决方案。

云风 提交于 June 28, 2013 05:37 PM | [固定链接](#)

## COMMENTS

@cloud, , , 多谢！

原来的代码我在lua5.1下能运行，现在修改后我将luaL\_traceback按原代码处理，会出现执行main.lua后segment fault错误，看来必须要转到5.2了。

初学lua，lua5.2的资料也不多，呵呵。

Posted by: cd | (11) [July 16, 2013 07:48 PM](#)

@cd

因为 Lua jit 已经实现了 luaL\_traceback

如果不用 jit, 请使用 lua 5.2

---

Posted by: Cloud | (10) [July 16, 2013 06:54 PM](#)

@cloud, 你移除了luaL\_traceback后, service\_lua.c中的traceback函数中还在使用这个, , , 执行的时候错误。

---

Posted by: cd | (9) [July 16, 2013 05:11 PM](#)

旧楼层太长, 这里就近提个小bug, static int HARBOR = 0; --》应改为无符号型, 否则右移导致算术移位, 非逻辑移位, 具体表现在当设置harbor>127时, 右移出负数, 远程消息会丢失

---

Posted by: wesom | (8) [July 13, 2013 05:42 PM](#)

@fu122zh

Module 可以只有 init 函数, 没有 create 函数。不注册 create 函数时, 视为一定 create 成功。

---

Posted by: Cloud | (7) [July 6, 2013 12:47 AM](#)

skynet\_module\_instance\_create函数调用失败返回(void\*)(intptr\_t)(~0), 这个不是0吧, 但调用skynet\_context\_new判断是否失败时是if (inst == NULL)

---

Posted by: fu122zh | (6) [July 5, 2013 06:21 PM](#)

@wesom

谢谢。这个不是新改出的问题, 是以前就有这种可能。

应该把 dispatch 调用拿到 skynet.start 之外, 保证在启动后就注册。

我已经修改过来了。

---

Posted by: Cloud | (5) [July 2, 2013 09:12 PM](#)

这个bug修复的有问题, 表现如下:

```
[1000007] launch snlua launcher
```

```
[1000008] launch snlua main
```

```
Server start
```

```
[1000009] launch socket 128
```

```
Unknown request : snlua service_mgr
```

```
[1000007] lua call [1000008 to :1000007 : 2 msgsz = 17] error : ./lualib/skynet.lua:279: Can't dispatch type text
:
```

```
stack traceback:
```

```
[C]: in function 'error'
```

```
./lualib/skynet.lua:279: in function <./lualib/skynet.lua:255>
```

模块的dispatch还没注册, 即脚本还未加载, .launcher却已经开始处理消息, 调用回调, 发现text对应的func为nil

---

Posted by: wesom | (4) [July 2, 2013 06:23 PM](#)

@wesom

谢谢 改好了

@cd

认证放在 watchdog 也不算复杂. 可以单独做一个服务. 由 watchdog 提出请求.

本质上是一个排队认证服务.

---

Posted by: Cloud | (3) [July 2, 2013 10:53 AM](#)

针对以上的问题, 个人想法:

1、agent、client数量众多且需求功能稳定, luaVM初始速度跟不上的话, 是否用c写这两个服务会更好?

2、即便用c写了，如果仍然跟不上要求，是否提前生成一批均量的agent、client，用资源池的办法解决，空间换时间。

3、认证消息放在watchdog处理容易将watchdog复杂化，剥离确实从逻辑上更清晰。

Posted by: cd | (2) [July 1, 2013 02:26 PM](#)

[1000008] lua call [0 to :1000008 : 1 msgsz = 0] error : ./lualib/skynet.lua:58: ./lualib/skynet.lua:176: Invalid name .launcher

stack traceback:

[C]: in function 'send'

./lualib/skynet.lua:176: in function 'send'

./lualib/skynet.lua:154: in function 'exit'

./service/main.lua:15: in function 'start'

two-step launch lua service --> 这个改动会导致启动时可能出现上述bug，main.lua在调用exit时报错，因为launcher的first message还来不及处理

Posted by: wesom | (1) [July 1, 2013 01:47 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)