

云风的 BLOG

思绪来得快去得也快，偶尔会在这里停留

[« 重写了 skynet 中的 socket 库](#) | [返回首页](#) | [skynet 下的用户登陆问题 »](#)

Hive, Lua 的 actor 模型

上个周末我一直在想，经过一年多在 [skynet](#) 上的开发，我已经有许多相关经验了。如果没有早期 [erlang](#) 版本的历史包袱以及刚开始设计 [skynet](#) 时的经验不足，去掉那些不必要的特性后的 [skynet](#) 应该是怎样的。

一个精简过代码的 [skynet](#) 不需要支持 [Lua](#) 之外的语言和通讯协议。如果某个服务的性能很关键，那么可以用 [C](#) 编写一个 [Lua](#) 库，只让 [Lua](#) 做消息分发。如果需要发送自定义协议的消息，可以把这个消息打包为一个 [C](#) 结构，然后把 [C](#) 结构指针编码在发送的消息中。

[skynet](#) 的内部控制指令全部可以移到一个系统服务中，用 [Lua](#) 编写。

跨机支持不是必要的。如果需要在多个进程/机器上运行多份协同工作，可以通过编写一个跨机通讯的服务来完成。虽然会增加一个间接层使跨进程通讯代价更大，但是可以简化许多代码。

广播也不是基础设施，直接用循环发送复制的消息即可。为了必要过大的消息在广播过程中反复拷贝，可以把需要广播的消息先打包为 [C](#) 对象，然后仅广播这个 [C](#) 对象的指针即可。

基于精简设计的理念，我花了两天时间重新实现了 [skynet](#)，并赋予它新的名字 [Hive](#)。这次一共不到 2000 行代码，其中 [C](#) 代码不到 1400 行，可以说达到我最初精简代码的目的。

对于不了解 [skynet](#) 的同学来说，最简单介绍 [hive](#) 的方法是，这是一个精简版的 [erlang](#)，用 [Lua](#) 实现的 actor 模型。

你可以用标准的 [Lua 5.2](#) 解释器包含 [hive](#) 模块，并用 `hive.start` 启动它即可。

`main.lua` 这个文件会被首先运行，并启动在一个独立的 [Lua state](#) 中，这里把这个独立的 [state](#) 称为 [cell](#)。同时，系统 [system cell](#) 一定存在，它可以向其他 [cell](#) 提供 [timer](#) 等基础服务。

[hive](#) 由很多 [cell](#) 构成，每个都是独立的 [lua state](#)，并采用多线程并行工作。线程的总数可以在启动时设定，[cell](#) 的个数可以远大于线程数，它们将完全均匀的分配 [cpu](#) 资源。

每个 [cell](#) 都是一个消息收发器，可以处理别的 [cell](#) 发送过来的消息。处理消息可以是严格的请求回应模式，也可以只投递消息而不期待回应。每条消息都由一个独立的 [coroutine](#) 来处理。

目前 `main.lua` 演示了如何启动新的 `cell pingpong.lua`，并给它发送消息。

感兴趣的同学可以在 [github](#) 上取得全部代码，我列了一个长长的 [todo list](#)，大部分工作是从 [skynet](#) 移植过来。这不是一个工作项目，所以不保证最近有精力全部完成它们。如果你有兴趣，可以帮助我一起来完成。

云风 提交于 June 26, 2013 01:15 PM | [固定链接](#)

COMMENTS

之前收到了你的[skynet](#)以及[gevent](#)等的启发，前段时间写了[Luactor](#)，非常简单的纯[lua](#)的单线程的Actor Model实现。和你面向游戏不同，我主要用在嵌入式开发里。

<https://github.com/xfguo/luactor>

Posted by: Xiongfei Guo | (15) [July 9, 2014 01:00 PM](#)

cloud:
udp也需要支持吧

Posted by: test | (14) [July 3, 2013 05:46 PM](#)

[hadoop](#)有一个组件就叫[Hive](#).
(<http://hive.apache.org>)

Posted by: Lee | (13) [July 2, 2013 01:48 PM](#)

@footer

目前是公平调度. 实现原则上不要让一个服务占据过久的时间片.

公平调度原则不可能让任何一个服务饿死.

Posted by: Cloud | (12) [June 28, 2013 02:06 PM](#)

想知道你们是怎样解决服务与服务之间竞争资源的问题. 如果是在一台机器下启动很多个不同类型的服务, 各服务之间的消息吞吐量存在很大差异, 这样可能会导致很多流程阻塞在一个点上.

有没有考虑过加个调度的优先级

Posted by: footer | (11) [June 28, 2013 12:46 AM](#)

@sw

调度器必须和消息处理在不同的 `coroutine` 里才可以被调度. 所以不同的消息一定在不同的 `coroutine` 中.

所谓等到 `request` 时再创建已经晚了. 如果推迟到 `request` 再创建, 那么 `request` 前的执行序和 `request` 之后的就在不同的 `coroutine` 中, 这样就得用 `callback` 函数才能工作起来.

lua 做调度工作并不慢, 进出 lua/C 边界, 以及边界上的数据交换往往才是性能热点.

没有 `gc` 和 `coroutine` 支持的 C/C++, 编写调度器也没有什么优势可言.

挂起执行序的也不一定是 `rpc` 调用, 也有自身的内部逻辑.

Posted by: Cloud | (10) [June 27, 2013 02:42 PM](#)

做游戏, 其实很怕想多了.

Posted by: 键盘老农 | (9) [June 27, 2013 11:51 AM](#)

做游戏, 其实很怕想多了.

Posted by: 键盘老农 | (8) [June 27, 2013 11:45 AM](#)

像底层`rpc`消息分发这样的功能, 并不经常变动, 用lua来实现有什么优势么? 性能上多少都会有些折扣吧.

“每条消息都由一个独立的 `coroutine` 来处理”这个有什么特别的需求么? 如果是请求回应为了简化回调模型, 在 `request` 的时候再创建协程是不是稍微节约一些呢.

比如相对用c/c++做`rpc`底层, lua处理消息逻辑. `rpc`请求的时候`yield coroutine`, 当底层`rpc`回应的时候c/c++再 `resume lua coroutine`. 用lua处理的优势在哪里呢?

Posted by: sw | (7) [June 27, 2013 08:52 AM](#)

我花了两天"实现"重新实现了 `skynet`, 实现->时间.

Posted by: joewan | (6) [June 26, 2013 05:25 PM](#)

写篇blog介绍一下你的开发环境和工具吧

Posted by: tiger | (5) [June 26, 2013 02:58 PM](#)

@cloud

你在什么环境下开发的?

Cygwin?

Posted by: tiger | (4) [June 26, 2013 02:57 PM](#)

@cd

你说那些在 `todo list` 里。

`skynet` 现在在我们的工作项目里用，围绕它的代码数倍于开源的核心部分，当然不会停。

Posted by: Cloud | (3) [June 26, 2013 02:32 PM](#)

@cloud

不知道hive的适用因应用是什么？一些基础的功能，如`socket`、`gate`、广播、跨机这些，是一个`server`框架必须的。看来，hive有可能成为`skynet`的一个新的整理和优化。

一直在学习你的`skynet`代码，希望不要停啊。

Posted by: cd | (2) [June 26, 2013 02:22 PM](#)

云风有没有考虑过在项目用`Golang`？感觉很适合游戏服务器开发。

Posted by: Jee | (1) [June 26, 2013 02:16 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）