

讲师介绍



Hash QQ: 805921455

从事Java软件开发近十年。
前新浪支付核心成员、
咪咕视讯(中国移动)项目经理、
对分布式架构、高性能编程有深入的研究。

明天，你一定会感谢今天奋力拼搏的自己

让你的Nginx更强大， 通过Lua拓展Nginx

分布式高并发—负载均衡

目录

课程安排



01

Lua脚本简述

像Lua脚本一样快



02

Ng i n x增加Lua执行模块

ngx_lua相关概念、安装
使用



03

打造高性能后端接口

强强联合，如何玩转
Ng i n x+Lua+Red i s?



04

功能实战

实现一个高性能的商品
页面

01

Lua脚本简述

Lua脚本语言



Lua是一种功能强大，高效，轻量级，可嵌入的脚本语言，非常容易嵌入到我们应用程序中。

Lua是动态类型的，通过使用基于寄存器的虚拟机解释字节码来运行，并具有增量垃圾收集的自动内存管理，使其成为配置，脚本和快速原型设计的理想选择。

Lua旨在成为一种轻量级可嵌入脚本语言。它用于各种应用程序，从游戏到Web应用程序和图像处理。

应用场景

1. 许多工业应用（例如， Adobe的Photoshop Lightroom）
2. 重点是嵌入式系统（例如， 巴西的数字电视的 Ginga中间件）
3. Lua目前 是游戏中领先的脚本语言，在游戏（例如， 魔兽世界和愤怒的小鸟）中使用。

官网：<https://www.lua.org>

Lua脚本特点

快，Lua是解释脚本语言领域中最快的语言

Lua是便携式的，在具有标准C编译器的所有平台中构建成开箱即用，可运行各种Unix和Windows，移动设备（运行Android，iOS，BREW，Symbian，Windows Phone），嵌入式微处理器（如ARM和Rabbit，适用于Lego MindStorms等应用程序），IBM大型机等。

可嵌入，Lua是一种快速语言引擎，占用空间小，可以轻松嵌入到应用程序中。

简单而强大，提供实现功能的*元机制*，而不是直接在语言中提供大量功能。Lua不是纯粹的面向对象语言，提供了实现类和继承的元机制。

小，源包含大约24000行C，包含源代码和文档的Lua 5.3.5的tar包需要297K压缩和1.2M未压缩。

免费，MIT许可证，可以用于任何目的，包括商业目的，完全免费。

开始使用Lua

安装Lua

```
curl -R -O http://www.lua.org/ftp/lua-5.3.5.tar.gz  
tar zxf lua-5.3.5.tar.gz  
cd lua-5.3.5  
make linux test
```

Hello World

```
$ lua  
Lua 5.1.4 Copyright (C) 1994-2008 Lua.org, PUC-Rio  
> print("hello, world")  
hello, world
```

一个小例子

新建hello.lua文件，内容如下

```
-- defines a factorial function
function fact (n)
    if n == 0 then
        return 1
    else
        return n * fact(n-1)
    end
end

print("enter a number:")
a = io.read("*number")      -- read a number
print(fact(a))
```

运行示例hello.lua文件

```
$ lua hello.lua
enter a number:
5
120      # 120=5*4*3*2*1
```

语法比Java要简略，缩进对齐，无大括号分号

Lua实现的阶乘代码示例

02

Ng i n x增加Lua执行模块

给老虎插上翅膀—Nginx嵌入Lua脚本语言

ngx_lua模块

不鼓励自己用Nginx构建这个模块，自己构建需要注意，Nginx，LuaJIT和OpenSSL官方发行版具有各种限制和长期存在的错误，这些错误可能会导致某些模块的功能被禁用，无法正常运行或运行速度变慢。如何编译，[参考《Nginx嵌入Lua脚本语言操作》手册](#)。

使用ngx_lua模块相当于使用Openresty，建议用官方版本OpenResty，它做了大量优化、集成了常用的第三方插件。

OpenResty

OpenResty是一款基于NGINX 和 LuaJIT 的 Web 平台。



OpenResty内部集成了大量精良的 Lua 库、第三方模块以及大多数的依赖项。用于方便地搭建能够处理超高并发、扩展性极高的动态 Web 应用、Web 服务和动态网关。

OpenResty的安装参考《OpenResty安装操作》，比构建ngx_lua模块简单很多

ngx_lua支持的指令

认识ngx_lua中的指令

指令分为配置指令、控制指令。

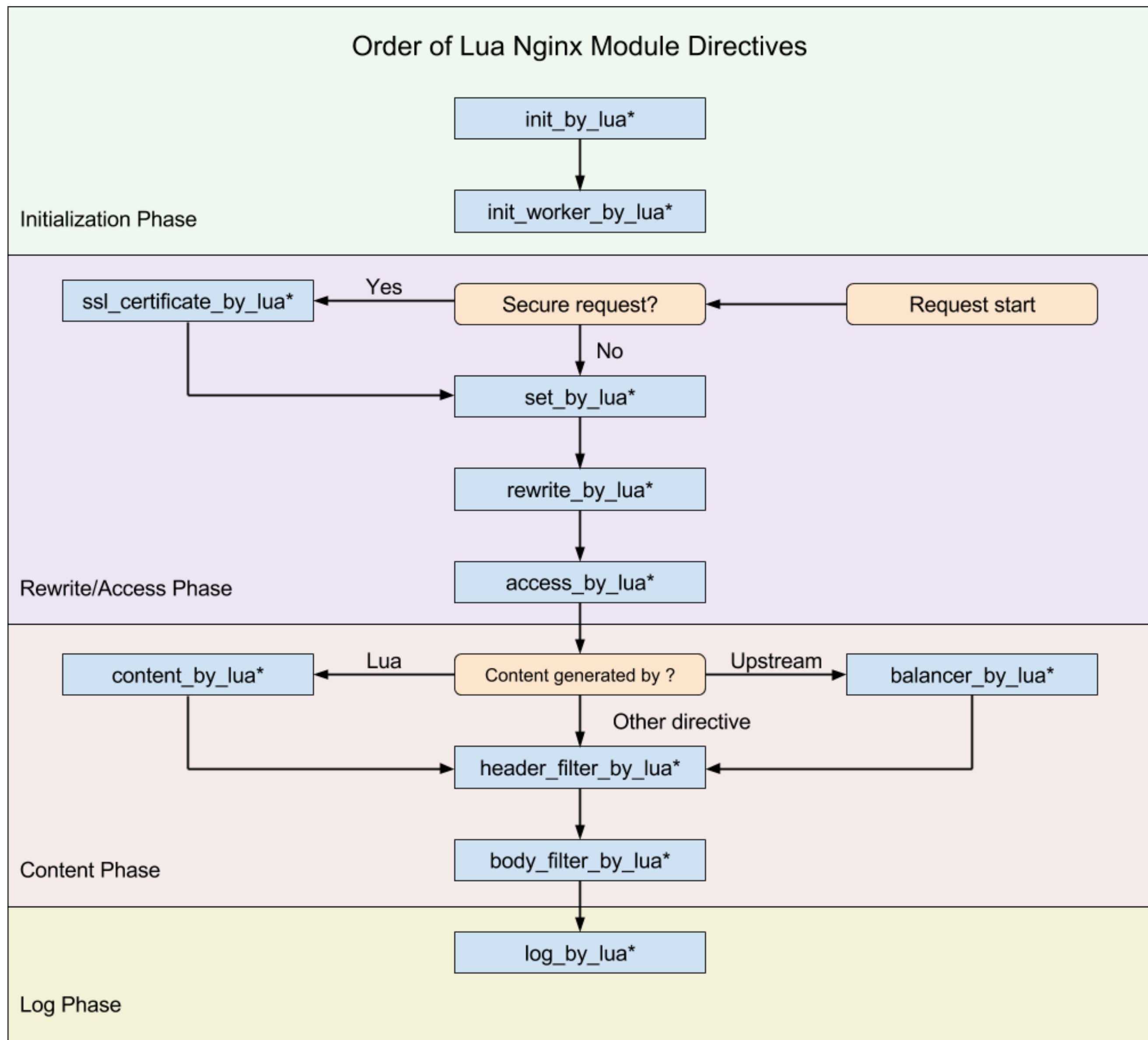
控制指令分为两种方式：

lua脚本块*_by_lua_block

lua脚本文件*_by_lua_file

Lua脚本执行顺序，如右图从上至下：初始化、重写/访问、内容处理、日志输出四个阶段。

内容处理可以由上游服务或者Lua脚本语言来处理，用Lua脚本就大有可为了。



ngx_lua演示

在OpenResty中演示ngx_lua的指令

内容，参考《nginx_lua_directives.conf》文件



打造高性能后端接口

OpenResty Redis模块

OpenResty中默认嵌入了下列模块，包含Redis，所以要使用Redis非常简单。

- lua-resty-memcached
- lua-resty-mysql
- lua-resty-redis
- lua-resty-dns
- lua-resty-upload
- lua-resty-websocket
- lua-resty-lock
- lua-resty-logger-socket
- lua-resty-lrucache
- lua-resty-string
- ngx_memc
- ngx_postgres
- ngx_redis2
- ngx_redis
- ngx_proxy
- ngx_fastcgi

-- 引入Redis模块

```
local redis = require("resty.redis")
```

--创建实例

```
local red = redis:new()
```

--建立连接

```
local ok, err = red:connect("127.0.0.1", 6379)
```

--调用API进行处理

```
ok, err = red:set("msg", "hello world")
```

```
local resp, err = red:get("msg")
```

Lua中使用Redis代码示例

Nginx非阻塞与Lua协程的绝配

服务器有个设计原则：永远不能阻塞。

nginx作为非常优秀的服务器，这点发挥的非常极致。在nginx里有很多的体现异步的地方。

Lua嵌入会阻塞吗？

```
-- foo本身是个协程，由nginx调用
function foo()
    -- 此处由nginx发起db的连接请求，因为异步这里先暂停协程
    users = db.query('users', {name: name});
    -- 当请求有响应得到处理时，继续执行协程，这些才运行
    deal_with(users);
end
```

通过nginx的异步机制和lua的协程，很容易实现这种同步写法，异步实现的模型，但对开发人员而言，不用关心内部做了什么。

Lua协程

什么是协程？

Lua协程，称为**协作式多线程**。Lua中的协程代表一个独立的执行线程，具有自己的堆栈、自己的局部变量、PC计数器和其自己的指令指针；与java线程不同的是，协程只通过显式调用yield函数来暂停其执行。同一时间，多协程只运行其中一个，java多线程则可以运行多个。

协程可以处于三种不同状态之一：**暂停，运行和死亡**

使用Lua协程

`coroutine.create(f)`

创建一个新的协同程序并返回一个类型为thread的对象，并不启动协程。

`coroutine.resume(co)`

开始或继续执行协同程序co

`coroutine.yield`

暂停执行调用协程，让出CPU

`coroutine.wrap(f)`

类似create方法，创建一个协程但它不是返回协程本身，而是返回一个函数，当被调用时，它恢复协程

Lua协程示例

同步的写法，异步的实现

```
co = coroutine.create(function ()  
    for i=1,10 do  
        print("co", i)  
        coroutine.yield()  
    end  
end)
```

```
coroutine.resume(co)  -- 输出 co 1  
coroutine.resume(co)  -- 输出 co 2  
coroutine.resume(co)  -- 输出 co 3
```



功能实战

商品页面需求

商品页主要包括商品基本信息、商品介绍、其他信息。

商品基本信息：

基本信息、图片列表、颜色/尺码关系、扩展属性、规格参数、包装清单、售后保障等

其他信息：

分类、品牌、店铺【第三方卖家】、店内分类【第三方卖家】、同类相关品牌

亿级别的商品数量，每次动态获取上面内容进行模板拼装，数据来源多到造成性能无法满足要求，而且页面需要动态支持多套模板，怎么处理？



商品详情 累计评价 商品推荐

商品描述

创意造型 浓浓文艺气息 闲暇时光 与好友分享

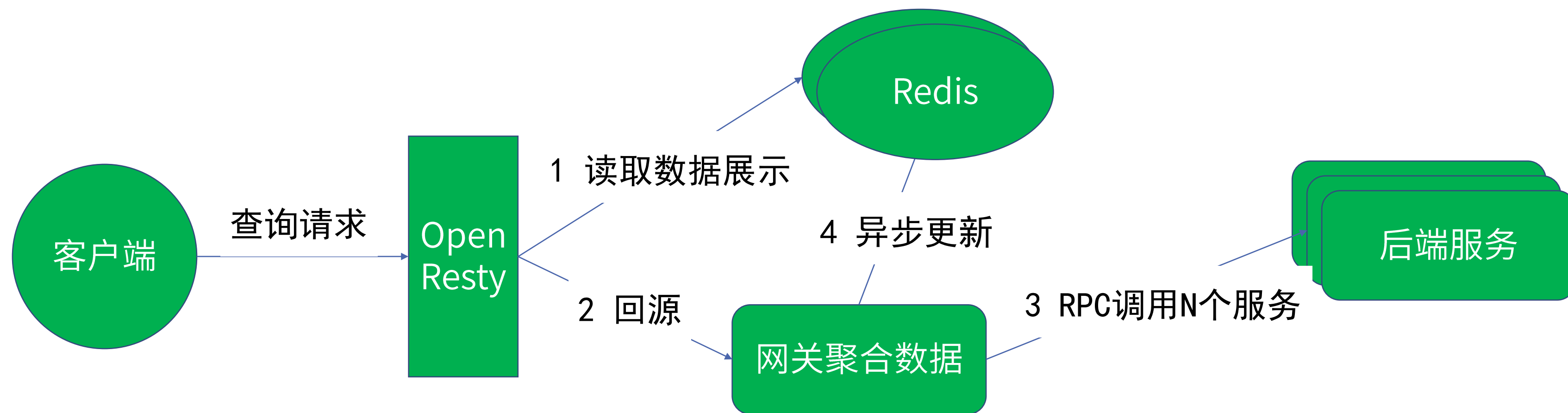
整体款式



商品页面功能设计

可以通过Nginx的API服务功能来实现

将需要的数据准备好，通过Lua访问Redis获取得，再由Lua生成json数据，交给Lua模板解析器，通过Lua模板解析器，来完成模板数据的动态生成。



商品页面功能实现

程序实现

`goods/info?goodsId=234`

Lua脚本实现

`commond.lua、goods_template.lua`

Nginx配置文件实现

`nginx_goods_template.conf`

总结

初始Lua脚本语言

脚本语言的特点

给老虎插上翅膀

Nginx+Lua、ngx_lua模块指令、指令执行顺序及阶段

高性能接口

ngx_lua中的第三方模块（Redis、MySQL、HTTP、Template）

为什么Nginx与Lua协程天生绝配？

商品页面实战

redis、nginx_lua、lua-resty-redis、lua-resty-template

谢谢观看