

# OpenResty的现状、趋势、使用及学习方法

温铭，奇虎 360 企业安全服务端架构师，OpenResty 社区咨询委员会成员。一直在互联网安全公司从事服务端的开发和架构工作，致力于用互联网技术帮助企业提高安全防护。曾负责开发过\*\*\*云查杀和反钓鱼系统。

## 1. OpenResty 是什么，适合什么场景下使用

和大部分知名开源软件诞生在欧美国家不同，OpenResty 自身和依赖的主要组件都是金砖国家的开发者发明的，这点还挺有意思。

Nginx 是俄罗斯人发明的，Lua 是巴西几个教授发明的，中国人章亦春把 LuaJIT VM 嵌入到 Nginx 中，实现了 OpenResty 这个高性能服务端解决方案。

通过 OpenResty，你可以把 nginx 的各种功能进行自由拼接，更重要的是，开发门槛并不高，这一切都是用强大轻巧的 Lua 语言来操控。

它主要的使用场景主要是：

在 Lua 中揉和和处理各种不同的 nginx 上游输出 (Proxy, Postgres, Redis, Memcached 等)

在请求真正到达上游服务之前，Lua 可以随心所欲的做复杂的访问控制和安全检测

随心所欲的操控响应头里面的信息

从外部存储服务（比如 Redis, Memcached, MySQL, Postgres）中获取后端信息，并用这些信息来实时选择哪一个后端来完成业务访问

在内容 **handler** 中随意编写复杂的 web 应用，使用 同步但依然非阻塞 的方式，访问后端数据库和其他存储

在 **rewrite** 阶段，通过 Lua 完成非常复杂的 **URL** dispatch

用 Lua 可以为 nginx 子请求和任意 location，实现高级缓存机制

组织 OpenResty 技术大会之前，我一直认为自己是一个孤独的 OpenResty 使

用者，觉得自己在使用一个冷门的技术。

虽然大家都听说过 OpenResty 或者 ngx\_lua，但感觉用在生产环境中使用的却少之又少，除了几个 CDN 公司外，好像没有听说过哪家知名互联网公司在用。而 CDN 行业之所以使用，很多是受到 cloudflare 技术栈的影响，OpenResty 的作者也在国外这家 CDN 公司。

但办完这个大会，我发现使用者真的挺多，奇虎360的所有服务端团队都在使用，京东、百度、魅族、知乎、优酷、新浪这些互联网公司都在使用。有用来写 WAF、有做 CDN 调度、有做广告系统、消息推送系统，还有像我们部门一样，用作 API server 的。有些还用在非常关键的业务上，比如开涛在高可用架构分享的京东商品详情页，是我知道的 ngx\_lua 最大规模的应用。

## 2. 奇虎企业安全服务端技术选型的标准

先说下 3 年多前做架构选型的时候，我为什么会选择 OpenResty？

其实架构如何设计并不重要，因为每家公司，每个团队，他们的公司文化和技术背景各不相同，生搬硬套会适得其反。重要的是当初为什么这么选择，中途为什么调整。

我们的产品要求单机上面，服务端提供高性能的 API 接口，QPS 至少过万，未来需要支撑到 10 万。我们并没有急于去使用 PHP、Python 或者其他语言来实现功能，而是先勾勒出一个理想化的技术模型。

这个模型应该具备：

非阻塞的访问网络IO。在连接 MySQL、Redis 和发起 HTTP 请求时，工作进程不能傻傻的等待网络IO的返回，而是需要支持事件驱动，用协程的方式让 CPU 资源更有效的去处理其他请求。很多语言并不具备这样的能力和周边库。

有完备的缓存机制。不仅需要支持 Redis、Memcached 等外部缓存，也应该在自己的进程内有缓存系统。我们希望大部分的请求都能在一个进程中得到数据并返回，这样是最高效的方法，一旦有了网络IO和进程间的交互，性能就会受到很大影响。

同步的写代码逻辑，不要让开发者感知到回调和异步。这个也很重要，程序员也是人，代码应该更符合人的思维习惯，显式的回调和异步关键字，会打断思路，也给调试带来困难。

最好是站在巨人肩上，基于成熟的技术上搭建。采用一门全新诞生的语言和技术，需要经历语言自身发展期频繁调整的阵痛，还可能站错队。

不仅支持 Linux 平台，还需要支持 Windows 平台，这个是我们产品很特别的需求，很多中小企业用户还是习惯 Windows 的操作，不具备 Linux 的维护能力。

基于以上几点的考虑，考察了当时的一些方案，选择了 OpenResty。

首先，它最大的特点就是用同步的代码逻辑实现非阻塞的调用，其次它有单进程内的 LRU cache 和进程间的 share DICT cache，而且它是揉合 nginx 和 LuaJIT 而产生的。而且 nginx 有 Windows 版本，虽然有非常多的限制，但这些限制都是可以解决的，nginx 官方 Windows 版本中不支持的特性，我们开源出来的版本都解决了。

第一次看到这样的方案，我觉得它肯定会颠覆高性能服务端的开发。为什么呢？在我之前的公司里，每天会有近百亿次的查询请求，而服务器只用了十台。

我们采用了 nginx C 模块 + 内置在 nginx 中的 K-V 数据库（自己开发的），来实现所有的业务逻辑，达到这个目标。听上去很简单，但是过程非常艰辛，两三个十几年工作经验的大牛做了一年多才稳定下来。绝大部分开发能力不足，只能望尘莫及。而且后续的调试和维护，也会花费不少精力。

但是 OpenResty 的出现改变了这一切，OpenResty 非常的 pythonic，适合人类的正常思维。新手经过一两个月的学习，做出来的 API，就可以达到 nginx C 模块的性能，而且代码量大大减少，也方便调试。

### 3. 以奇虎和新浪为例，如何在项目中引入新技术

技术选型只是第一步，如何才能在一个产品或者项目中引入 OpenResty 这个新的技术呢？我拿奇虎企业安全和新浪移动这两家公司真实发生的案例给大家看看。我和新浪移动的周晶，都是在一个有成熟产品的部门，用一两个人的力量，把一个新技术，替换掉了原有的技术架构。但由于企业产品和个人产品的不同，方法有很大的不一样。

先说我所在奇虎企业安全。我在 2012 年初加入这个部门，当时产品主打免费，目标用户是小企业。所以架构设计上面，只考虑了几十点、几百点的终端请求，使用了非常强绑定的 Windows 平台技术，而且倾向于不用开源软件，自己新做一个更适合自己的框架。包括自己用 C++ 开发的 Web

server，自己写的 PHP 路由和框架，数据存储在 sqlite 里面。

我帮忙修改了两个月 PHP 的 bug，看明白了技术架构的思路之后，就去新开的一个产品线了。这是一个实验性的产品，主要面对央企和专用网，一个网络中有上百万的终端。

刚开始没有什么人关注，我就直接采用了 Linux + OpenResty + Redis + Postgres 的开源组件，性能测试甩之前的N条街。后面这个实验性的产品，和之前的产品，合并为一个产品，技术上面就割裂为两套架构。老功能用老架构，新功能用新架构。

随着越来越多大用户的增加，原有的技术架构开始捉襟见肘，技术债务越积压越多。随着用户的抱怨，sqlite 被抛弃，全面换成 Postgres。但对于自己开发的框架还是有些敝帚自珍。

期间通过对比测试、OpenResty 培训还有多次用户性能问题排查，让开发同学们都知道这门技术的优势。快被加班压垮的开发同学，逐渐开始选择使用 OpenResty 而不是自研的框架，来进行新功能的开发，以及旧功能的迁移，来避免加班。

在产品重构的时候，之前自研的服务端框架被完全抛弃，服务端开发的同学从 8、9 个人减少到 3 个人。在新技术的引入过程中，我们没有采用强制的举措，因为企业产品需要稳定，用户处部署的版本更新很慢。

而新浪移动周晶的实践，对大家更有参考意义。新浪移动最开始是基于 Apache，用 PHP 来处理用户请求。Apache 是同步多进程模型，在并发请求不多的情况下没有问题。

但是总是会有突发新闻，比如马航失联、文章×××等，突发的高流量把后台压垮了几次。而且可以预见世界杯的流量也会很大，所以周晶花几个月时间，用 nginx 替换了 Apache，使用 nginx 的 fast\_cgi\_cache，QPS 提升了一个数量级。

新浪移动后台的接口都是使用 PHP 来实现的，在高并发下有些力不从心。而 nginx 简单的缓存虽然能满足性能，但不能满足业务精细化和数据一致性的要求，需要找 PHP 之外的解决方案，前提是让 PHP 的开发能够舒适的使用。node.js 的回调地狱、Go 的调试不方便，都是一个阻碍。

他们最后选择了 OpenResty，而且基于 OpenResty 开源了一个 Web 框架 Vanilla（香草），模仿了 Yaf 的使用习惯，让 PHP 的开发更容易接受和上手。Vanilla 已经在新浪移动开始使用，一些核心业务，比如高清图和体育直播，正在向这个框架迁移中。

#### 4. 入门痛点，以及学习的正确方法

我和周晶的入门，都是自己摸着石头过河。当时除了 Python 社区「大妈」的那篇使用文章外，找不到其他的资料。

奇虎和新浪都用 OpenResty 成功替换了之前的技术，但问题还是挺明显，就是大家都认为自己是孤独的使用者，同事中基本没有人认同。在关键和支撑业务上，使用 OpenResty 有些不放心，都会在边缘业务上先做尝试和验证。

虽然 OpenResty 的性能做的很棒，比肩或者超过其他所有的高性能解决方案，但是担心没有学习资料、担心招不到人、担心没人交流，可能还担心作者章亦春哪天撂挑子不干了，这个项目就黄了。

高可用架构群里的各位都是架构师，是技术决策者，在引入一门新技术的时候，肯定会考虑到这些风险。比如小米科技马利超在高可用架构的分享，他们在抢购系统中曾经使用过 ngx\_lua，虽然性能满足需求，但是团队里面熟悉的人少，最后还是改成了 Go 语言实现。

如何解决这些担忧？社区是有过思考和讨论的，我们放在分享最后讲。先从一个尝试使用这门技术的开发者的角度看，OpenResty 不少基础工作没有完善，友好程度不够：

只能从源码安装，没有 apt-get、brew 等软件仓库安装方法；安装第三方库没有 PIP、NPM 之类的包管理工具，需要去先谷歌，然后拷贝代码文件到指定的目录下，才能 require 使用。

代码编写需要修改 nginx.conf 和对应的 lua 代码，即使是 hello world 也是如此。当然你可以把代码写在 nginx 的配置文件里面，但是生产环境肯定是要分离的。这种编写代码的方式，不像是一个编程语言，和常规的编程方式不同。

有独特的执行阶段概念，因为 OpenResty 是基于 nginx 的，所以也继承它的这种概念。你的代码逻辑，可能需要放在不同的阶段里面运行，才能获取你想要的预期。而这些阶段间信息如何传递，以及哪些 API 不能在某些阶段使用，就会经常拦住新手。

遇到问题只有邮件列表这一种方式来沟通，而邮件列表是被墙的。文档也只有英文版本，导致很多新手的问题无法被解决。

没有系统学习 OpenResty 的手段，大都是业务需要实现什么功能，就去文档和 API 里面去找。至于方式对不对，能不能优化，就不知道了。

而 Lua 语言自身也有一些特别的地方：

下标从 1 开始，这个是和其他编程语言很大的不同。

不区分 array 和 dict，会导致处理 json 的时候，无法区分 array 和 object。

默认全局变量，需要在所有变量前加 local，忘记的话，可能导致各种难查的 bug。

自带的字符串正则匹配规则和通常的 PCRE 不同，使用的话，学习成本较高。

Lua 标准库和周边库，都是阻塞的，需要自己甄别哪些可以和 OpenResty 搭配使用。新手很容易使用了阻塞的库，而导致性能急剧下降。

有没有好的入门方法？

我们团队正在做这方面的努力，尽量在现有的基础上，降低学习的门槛。

对于新手，可以看 StuQ 上面 OpenResty 的系列视频教程

(<http://www.stuq.org/course/detail/1015>)，我们计划有 4 季，分别是入门、进阶、实战和源码分析。现在第一季已经上线，第二季正在后期制作。看完前两季，基本上就可以在项目里面用了。

对于已经使用了 OpenResty 的开发者，我们把这两三年遇到的坑，都记录在 GitHub 的《OpenResty最佳实践》上面

(<https://github.com/moonbingbing/openresty-best-practices>)，大家可以当做 cookbook 来使用。

## 5. nginxScript 这样的尝试会替代 OpenResty 吗？

nginxScript 是今年 nginx 大会上，Nginx 官方推出的一个新的配置语言。它是模仿了 OpenResty 的做法，把 JavaScript VM 嵌入到 nginx 中，提供简单的 nginx 配置功能。

我们看下它的 hello world：

```
location / {
    js_run "
        var res;
        res = $r.response;
        res.status = 200;
        res.send('Hello World!');
        res.finish();
    ";
}
```

@51CTO博客

再对比下 OpenResty 的 hello world:

```
location / {
    content_by_lua_block {
        ngx.say("hello world")
    }
}
```

@51CTO博客

看上去差不多，只是 OpenResty 简洁一些。根据 nginx 官方的说明，nginxScript 只是想提供一种更方便配置 nginx 的方法，并不想取代 ngx\_lua。

考虑到 JavaScript 本身的流行和开发社区的强大，如果未来两三年它从一个简单的 nginx 配置语言，逐渐演变成类似 ngx\_lua 这样功能非常完备的开发语言，甚至替代 OpenResty 也是有可能的。

当然，这个前提是 OpenResty 停滞不前。现在 OpenResty 已经有的功能，和计划开发的功能，倾向于覆盖 nginx Plus 的功能。所以 nginx 和 OpenResty 之间，有一个良性的竞争关系，这是大家都乐意看到的。

## 6 未来重点解决的问题和新增特性

短期内的目标，是想降低入门的难度：

提供官方二进制发布包。类似于 **docker** 的安装方法，一行命令，下载一个 **sh** 脚本，增加一个源地址，不用手工解决依赖，不用源码编译，直接就可以试用。



而且会发布 **Windows** 的二进制包，方便这个平台的开发者本机做一些测试。

增加包管理。命令行工具叫 **iresty**，可以从 **iresty.org** 上面搜索、安装需要的 **lua resty** 库，避免找错库或者放错目录。

写一本书《**OpenResty** 编程》，这本书会成为官方的入门书籍，框架和关键内容由作者春哥直接操刀，我和社区的其他同学帮助一起完成。

做完上面3点，OpenResty 的入门难度会降低到和其他编程语言一样。

在功能上面，会增加很多激动人心的新特性：

支持 TCP 和 UDP 。Nginx 最新的 stream 子系统已经支持了 TCP，OpenResty 的 ngx\_stream\_lua 模块正在开发中，会拥有和现有的 nginx http modlue 相同的 lua API，所以很多应用和库，可以不加修改的运行在一个新的子系统上面。

更好的支持推送场景。增加 shared list 共享内存的队列，可以用于 worker 间的通讯；增加 semaphore 特性，用于 ngx\_lua 轻量级线程间的通讯。酷狗音乐的推送服务就是基于这些实现的，这些改动点会在这个月并入 master。可以邀请酷狗音乐的同学，来给大家详细分享下里面的细节。

建立一个开源的 WAF 平台。现在阿里云和 cloudflare 的 WAF 做的都很棒，经受住了很多实际的考验。但是都没有开源，我们希望最好的 WAF 是开源的，而且是基于 OpenResty 的。

在 OpenResty 中增加内存数据库。可以有持久化，或者就是全内存的，支持 SQL 的查询。这个也是出于极致性能的考虑，有时候我们还是需要使用 SQL 来做一些复杂的查询，但我不想使用那么重的关系型数据库，而且数据是可以丢失的。那么这个就可以排上用场。

实现 PHP、Python 等方言，让 PHP、Python 等程序员可以用自己喜欢的语言写 OpenResty 的代码，底层转换为 LuaJIT 的字节码。

春哥在 OpenResty 技术大会上说了非常多的新特性，包括 streaming RegEx 正则引擎等等，非常高端，我挑了几个我觉得有意思的做介绍。

## 7. 开源社区建设

OpenResty 诞生于 2011 年，大多数时间都是春哥主力在维护这个项目，当然也有很多开发者提交 feature 和 bugfix，但基本上算单打独斗。

社区有 github 和邮件列表，大部分还是提问的。春哥每天会花费很多的时间，来详细的回答各种基础问题。



今年新增了 QQ 群和微信群，QQ 群的质量很高，每天都有很多提问，非技术问题是被禁止的。而且还有了自己的技术大会，能给大家面对面交流的机会。

我们翻译了 ngx\_lua 的英文文档，能让大家更方便的查找资料；我们搭建了一个不用 × × × 就能访问的论坛：bbs.iresty.com，用作提问和知识积累的地方。后面会把谷歌邮件列表的内容同步过来。

只有上面这些是不够的，在 OpenResty 技术大会的第二天，我们召集了一个很小规模的闭门会议，决定成立 OpenResty 咨询委员会。

这个委员会，是以个人名字参加的，成员来自奇虎 360、新浪、又拍云、酷狗音乐等公司和社区的开发者，希望把国内社区的核心使用者和开发者团结在一起，促进 OpenResty 的发展。

同时，OpenResty 软件基金会也开始筹备工作，我们希望走规范的非盈利组织的模式，来保证 OpenResty 长期稳定发展。给开发者和使用者信心，敢于在关键业务上面使用 OpenResty。

## Q & A

1、请问 OpenResty 的定位是什么，从分享来看似乎全栈了？

定位主要是高性能，所有的新功能和优化，都是针对性能的。但是也有人拿来做页面，比如京东；也有人拿来替代 PHP 做 Web server，比如新浪。我觉得它越来越像一个独立的开发语言。

2、请问 Lua 是不是可以实现动态配置 location？比如动态切流量？

balancer\_by\_lua 可能是你需要的，你可以用 Lua 来定义自己的负载均衡器，可以在每个请求的级别上去定义，当前访问的后端的节点地址、端口，还可以定制很细力度的访问失败之后的重试策略。

3、OpenResty 是可以拿到 nginx 请求里面的所有信息？那是不是可以做一些更复杂的转发操作？能介绍一下 OpenResty 在 cdn 里面的应用场景吗？

可以看下 iresty.com 的分享，又拍的张聪非常详细的介绍了 OpenResty 在又拍 CDN 的使用。

4、OpenResty 是否修改了 nginx 的源码，还是和 nginx 完全可剥离的？  
Nginx 版本升级，OpenResty 也跟着升级吗？例如 nginx 修复漏洞 bug 等情况。

OpenResty 不修改 nginx 的源码，可以跟随 nginx 无痛升级。如果你觉得 OpenResty 升级慢了，你可以只拿 ngx\_lua 出来，当做 nginx 的一个模块来编译。实际上，OpenResty 在测试过程中，发现了很多 nginx 自身的 bug。

5、软 WAF nginx + Lua 是主流和未来方向么？

我觉得 WAF 应该基于 nginx，不管是性能还是流行程度。而 OpenResty 具有更灵活操控 nginx 的能力，所以我觉得 OpenResty 在 WAF 领域非常合适。cloudflare 的 WAF 就是基于 OpenResty。

6、看样子未来可能有各种 ngx\_xx，最有可能的是 js，不知道这方面有什么前沿的动向？

我们组在尝试把 PHP 嵌入到 nginx 中，当然性能肯定不如 LuaJIT，但是会方便很多 PHP 同学，有进展的话，我们会开源出来：)

7、OpenResty 目前看似乎是一个 proxy 的配置框架（糅合了 nginx + Lua），但以后的发展是什么样子？会不会以后更进一步，比如做一个 API gateway 之类的。

OpenResty 其实是希望大家忽略 nginx 的存在，直接使用 ngx\_lua 提供的 API 实现自己的业务逻辑。更像一门独立的开发语言，只不过底层使用 nginx 的网络库而已。你可以按照你的想法搭建任何好玩的服务端应用出来。