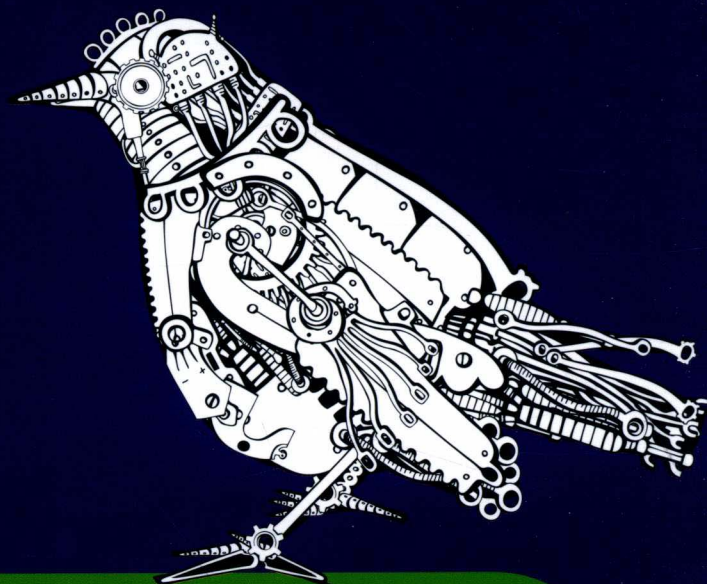


本书深入浅出地介绍了Nginx+Lua在实战场景中的各种使用技巧和方法，涉及Nginx配置、常用模块、缓存系统、日志分析、静态容灾、反向代理、爬虫、性能分析与优化等众多方面，掌握这些知识有助于提升你所开发的服务的性能。

Broadview
www.broadview.com.cn



Nginx 实战

基于Lua语言的配置、 开发与架构详解

王力 汤永全 著

 中国工信出版集团

 电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

Nginx 实战

基于Lua语言的配置、
开发与架构详解

王力 汤永全 著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书主要讲解了 Nginx 在反向代理和应用开发中的作用, 阅读本书可以了解 Nginx 在互联网开发中扮演的多个角色, 充分利用这些角色的各项功能有助于提升服务的整体性能。本书所介绍的大部分功能是通过 Nginx+Lua 进行开发和配置的, 但并不要求读者精通 Lua, 在必要的位置, 本书会对 Lua 进行选择性地讲解。涉及实战的内容会有配套源码, 方便读者学习和使用。

本书适合广大运维人员和开发人员学习, 对使用 Nginx 完成各种服务架构感兴趣的架构师也可以阅读本书。阅读本书需要有对 Nginx 的初级或中级配置经验。

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有, 侵权必究。

图书在版编目 (CIP) 数据

Nginx 实战: 基于 Lua 语言的配置、开发与架构详解 / 王力, 汤永全著. —北京: 电子工业出版社, 2019.3
ISBN 978-7-121-35460-1

I. ①N… II. ①王… ②汤… III. ①互联网络—网络服务器—程序设计 IV. ①TP368.5

中国版本图书馆 CIP 数据核字 (2018) 第 254599 号

策划编辑: 付 睿

责任编辑: 牛 勇

印 刷: 三河市华成印务有限公司

装 订: 三河市华成印务有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编: 100036

开 本: 787×980 1/16 印张: 21.5 字数: 465 千字

版 次: 2019 年 3 月第 1 版

印 次: 2019 年 3 月第 1 次印刷

印 数: 2500 册 定价: 79.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888, 88258888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式: (010) 51260888-819, faq@phei.com.cn。

前言

Nginx 自 2004 年发布第一个公开版本以来，就因其稳定性强、配置灵活、占用内存少、反向代理功能强大，而被越来越多的人喜爱和使用。随着人们对 Nginx 关注度的不断上升，Nginx 有了更多的使用场景，但在大多数公司中仍然只是扮演着反向代理的角色。

笔者在折 800 电商平台（以下简称折 800）工作多年，深感对一个电商平台来说，在成本控制和服务性能之间取得平衡是能够可持续发展的前提之一。与同类型且流量级别相近的公司相比，折 800 的计算机硬件成本要低很多，这主要得益于公司对软件技术的极致追求。在这样的背景下，我们热衷于研究 Nginx 反向代理等小众技术，并使用 Nginx 实现了大量功能，极大地提升了服务的性能和灵活度。

目前，Nginx 在国内的影响力还比较有限，因此很多开发人员并不了解 Nginx 的魅力，笔者写这本书的目的就是和大家分享 Nginx 的使用技巧，并一起来推广 Nginx。

市面上已经有一些介绍 Nginx 技术的书籍，但大多偏向于对 Nginx 配置和原理的讲解，还有一些书籍是基于 C 或 C++ 来介绍 Nginx 模块的。这样的书籍要求读者对 Nginx 的原理和源码有所了解，甚至还要有深厚的 C 语言或 C++ 语言的开发功底，这对大多数初学者来说要求太高了。

怎样才能做到既发挥出 Nginx 的威力，又尽量降低学习成本呢？笔者选择使用 OpenResty 的核心组件，因为 OpenResty 是 Nginx+Lua 的完美结合，它既能够实现 Nginx 的异步功能，又兼有 Lua 的易学优势，这样，在使用 Nginx 进行开发的过程中就不需要深入了解 Nginx 的原理了（如果已经掌握了 Nginx 原理当然更好）。所以，与市面上的大多数同类书籍不同，本书不会过多介绍 Nginx 配置和原理方面的内容，学习难度较低。

近几年来，OpenResty 的热度不断上升，各大互联网公司纷纷尝试使用 Nginx+Lua 的开发模式，在反向代理、网关系统、API 调度服务中都能看到这种开发模式的“身影”。希望本书可以让更多的开发人员了解 Nginx+Lua 的魅力，也让更多技术团队能够合理地使用 Nginx，降低硬件的投入成本，同时提升服务的性能。

本书由王力、汤永全著。全书内容共分 18 章，其中第 1~5 章介绍 Nginx 常见的配置方式，属于 Nginx 的入门知识；第 6~10 章对 Nginx+Lua 核心功能进行解读，是熟悉 Nginx+Lua 开发模式的必备知识；第 11~18 章是 Nginx+Lua 开发模式在实际业务中的实践应用，是灵活运用该模式的具体体现。

感谢彭赫、杨明翰、冯浩、刘凯、屈耀华对本书的支持；也感谢折 800 技术平台，在这里我得到了持续的历练和成长。同时，感谢电子工业出版社博文视点的编辑付睿和崔志伟，他们在本书的语言表述方面给了很多建议。

本书包含了作者的技术实践，如果你对本书内容有任何建议和疑惑，可以发电子邮件至 leehomewl@gmail.com。谢谢！

王力

2018 年 12 月

轻松注册成为博文视点社区用户（www.broadview.com.cn），扫码直达本书页面。

- **提交勘误：**您对书中内容的修改意见可在 [提交勘误](#) 处提交，若被采纳，将获赠博文视点社区积分（在您购买电子书时，积分可用来抵扣相应金额）。
- **交流互动：**在页面下方 [读者评论](#) 处留下您的疑问或观点，与我们和其他读者一同学习交流。

页面入口：<http://www.broadview.com.cn/35460>



目 录

第 1 章 Nginx 学前必知	1
1.1 HTTP 请求报文	1
1.2 HTTP 响应报文	2
1.3 安装 Nginx	2
1.4 支持 HTTPS	4
1.5 添加模块	4
1.6 小结	4
第 2 章 基础配置	5
2.1 Nginx 指令和指令块	5
2.2 Nginx 基本配置说明	6
2.2.1 main 配置	6
2.2.2 与客户端有关的配置	7
2.2.3 server 块	7
2.2.4 location 块	8
2.3 include 的使用	9
2.4 常见配置	9
2.4.1 常见配置注解	10
2.4.2 常见配置实战技巧	11
2.5 内置变量	13
2.5.1 常见内置变量	13
2.5.2 常见内置变量实战技巧	15
2.6 小结	16

第 3 章 强化基础配置	17
3.1 牢记 Context	17
3.2 获取请求的 IP 地址	18
3.2.1 获取用户的真实 IP 地址	18
3.2.2 防止 IP 地址伪造	19
3.2.3 后端服务器对 IP 地址的需求	19
3.3 管理请求的行为	20
3.3.1 限制 IP 地址的访问	20
3.3.2 auth 身份验证	21
3.3.3 利用 LDAP 服务加强安全	22
3.3.4 satisfy 二选一的访问限制功能	23
3.4 proxy 代理	23
3.4.1 proxy_pass 请求代理规则	24
3.4.2 减少后端服务器的网络开销	24
3.4.3 控制请求头和请求体	25
3.4.4 控制请求和后端服务器的交互时间	26
3.5 upstream 使用手册	26
3.5.1 代理多台服务器	27
3.5.2 故障转移	28
3.5.3 负载均衡	29
3.5.4 通过 hash 分片提升缓存命中率	29
3.5.5 利用长连接提升性能	30
3.5.6 利用 resolver 加速对内部域名的访问	31
3.6 rewrite 使用手册	32
3.6.1 内部重定向	32
3.6.2 域名跳转	33
3.6.3 跳转 POST 请求	34
3.6.4 设置变量的值	34
3.7 限速白名单	35
3.8 日志	36
3.8.1 记录自定义变量	36
3.8.2 日志格式规范	36
3.8.3 日志存储	37

3.9 HTTP 执行阶段	38
3.10 小结	39
第 4 章 常用模块精解	40
4.1 定制 HTTP 头信息	40
4.1.1 使用 ngx_http_headers_module 设置响应头	40
4.1.2 使用 headers-more-nginx 控制请求头和响应头	43
4.2 第三方模块 set-misc-nginx	45
4.2.1 设置变量	46
4.2.2 防止 SQL 注入	46
4.2.3 字符串非转义和转义	47
4.2.4 基于键值的集群分片	48
4.2.5 base 编码	48
4.2.6 md5 编码	50
4.2.7 生成随机数	50
4.2.8 本地时间的输出	52
4.2.9 实战经验	52
4.3 图片的处理	53
4.3.1 image_filter 图片处理	53
4.3.2 采用渐进式方式打开 JPEG 图片	55
4.3.3 WebP 格式	56
4.3.4 优化图片	56
4.3.5 实战经验：动态切图	58
4.4 TCP 和 UDP 代理	58
4.4.1 代理配置说明	58
4.4.2 DNS 服务的反向代理	62
4.4.3 MySQL 集群代理配置	62
4.4.4 实战经验	63
4.5 常用模块介绍	63
4.5.1 基于访问 IP 地址跳转到对应城市	63
4.5.2 修改响应内容	65
4.5.3 零像素文件的生成及其作用	66
4.5.4 图片的防盗链	67

4.6 小结	68
第 5 章 缓存系统	69
5.1 缓存配置说明	69
5.2 控制缓存有效期	71
5.3 性能优化	72
5.3.1 缓存未命中的最佳实践	72
5.3.2 横向扩展最佳实践	75
5.3.3 避免硬盘 I/O 阻塞	76
5.3.4 集群模式	77
5.4 高可用方案	77
5.5 proxy_cache 配置模板	78
5.6 小结	81
第 6 章 引入 Lua	82
6.1 引入 Lua 的原因	82
6.2 Lua 和 LuaJIT	83
6.3 环境搭建	83
6.4 Lua 的数据类型	84
6.4.1 类型说明	84
6.4.2 类型示例	85
6.5 表达式	89
6.5.1 算术运算符	89
6.5.2 关系运算符	90
6.5.3 逻辑运算符	91
6.5.4 字符串连接和字符串长度计算	92
6.5.5 运算符优先级	93
6.6 变量	93
6.6.1 全局变量	94
6.6.2 局部变量	94
6.6.3 变量赋值	94
6.7 流程控制	95
6.7.1 if-else	95
6.7.2 for 循环	96

6.7.3	while 循环	97
6.7.4	break 和 return	97
6.8	函数	98
6.8.1	函数格式	98
6.8.2	传参方式	99
6.8.3	函数的创建位置	100
6.9	模块	100
6.9.1	模块格式	101
6.9.2	加载模块	101
6.10	Lua 常见操作	102
6.10.1	操作 table	102
6.10.2	定义字符串	103
6.10.3	字符串连接	104
6.11	引入 Lua 的插曲	104
6.12	小结	105
第 7 章	Lua-Nginx-Module 常用指令	106
7.1	Nginx 和 OpenResty	106
7.2	安装 Ngx_Lua	107
7.3	牢记 Context	108
7.4	Hello World	108
7.5	避免 I/O 阻塞	109
7.6	定义模块搜索路径	109
7.6.1	定义 Lua 模块的搜索路径	109
7.6.2	定义 C 模块的搜索路径	110
7.7	读/写 Nginx 的内置变量	110
7.8	控制请求头	111
7.8.1	添加请求头	111
7.8.2	清除请求头	112
7.8.3	获取请求头	112
7.9	控制响应头	113
7.9.1	获取响应头	113
7.9.2	修改响应头	114

- 7.9.3 清除响应头116
- 7.10 读取请求体116
 - 7.10.1 强制获取请求体116
 - 7.10.2 用同步非阻塞方式获取请求体117
 - 7.10.3 使用场景示例118
 - 7.10.4 使用建议121
- 7.11 输出响应体121
 - 7.11.1 异步发送响应体121
 - 7.11.2 同步发送响应体122
- 7.12 正则表达式124
 - 7.12.1 单一捕获124
 - 7.12.2 全部捕获125
 - 7.12.3 更高效的匹配和捕获126
 - 7.12.4 替换数据128
 - 7.12.5 转义符号129
- 7.13 子请求130
 - 7.13.1 请求方法130
 - 7.13.2 单一子请求130
 - 7.13.3 并发子请求134
- 7.14 获取 Nginx 的环境变量135
 - 7.14.1 获取环境所在的模块135
 - 7.14.2 确认调试模式136
 - 7.14.3 获取 prefix 路径136
 - 7.14.4 获取 Nginx 的版本号136
 - 7.14.5 获取 configure 信息136
 - 7.14.6 获取 ngx_lua 的版本号137
 - 7.14.7 判断 worker 进程是否退出137
 - 7.14.8 获取 worker 进程的 ID137
 - 7.14.9 获取 worker 进程的数量137
- 7.15 定时任务138
 - 7.15.1 创建定时任务138
 - 7.15.2 性能优化140
 - 7.15.3 禁用的 Lua API141

7.16 常用指令	142
7.16.1 请求重定向	142
7.16.2 日志记录	144
7.16.3 请求中断处理	146
7.17 提升开发和测试效率	149
7.17.1 断开客户端连接	149
7.17.2 请求休眠	150
7.17.3 获取系统时间	150
7.17.4 编码与解码	152
7.17.5 防止 SQL 注入	154
7.17.6 判断是否为子请求	155
7.17.7 设置 MIME 类型	156
7.18 小结	156
第 8 章 Ngx_Lua 的执行阶段	157
8.1 init_by_lua_block	157
8.1.1 阶段说明	157
8.1.2 初始化配置	158
8.1.3 控制初始值	159
8.1.4 init_by_lua_file	160
8.1.5 可使用的 Lua API 指令	160
8.2 init_worker_by_lua_block	160
8.2.1 阶段说明	160
8.2.2 启动 Nginx 的定时任务	161
8.2.3 动态进行后端健康检查	162
8.3 set_by_lua_block	165
8.3.1 阶段说明	165
8.3.2 变量赋值	165
8.3.3 rewrite 阶段的混用模式	166
8.3.4 阻塞事件	167
8.3.5 被禁用的 Lua API 指令	167
8.4 rewrite_by_lua_block	168
8.4.1 阶段说明	168

8.4.2	利用 <code>rewrite_by_lua_no_postpone</code> 改变执行顺序	168
8.4.3	阶段控制	169
8.5	<code>access_by_lua_block</code>	169
8.5.1	阶段说明	169
8.5.2	利用 <code>access_by_lua_no_postpone</code> 改变执行顺序	170
8.5.3	阶段控制	170
8.5.4	动态配置黑白名单	170
8.6	<code>content_by_lua_block</code>	170
8.6.1	阶段说明	170
8.6.2	动态调整执行文件的路径	171
8.7	<code>balancer_by_lua_block</code>	171
8.7.1	阶段说明	171
8.7.2	被禁用的 Lua API 指令	172
8.8	<code>header_filter_by_lua_block</code>	172
8.8.1	阶段说明	172
8.8.2	被禁用的 Lua API 指令	173
8.9	<code>body_filter_by_lua_block</code>	173
8.9.1	阶段说明	173
8.9.2	控制响应体数据	173
8.9.3	被禁用的 Lua API 指令	175
8.10	<code>log_by_lua_block</code>	176
8.10.1	阶段说明	176
8.10.2	被禁用的 Lua API 指令	176
8.11	Lua 和 <code>ngx.ssl</code>	177
8.12	<code>Ngx_Lua</code> 执行阶段	177
8.13	小结	180
第 9 章	Nginx 与数据库的交互	181
9.1	安装 <code>cjson</code>	181
9.2	与 MySQL 交互	183
9.2.1	安装 <code>lua-resty-mysql</code> 模块	183
9.2.2	读取 MySQL 数据	183
9.2.3	执行多条 SQL 语句	187

9.2.4 防止 SQL 注入	189
9.3 与 Redis 交互	189
9.3.1 安装 lua-resty-redis	189
9.3.2 读/写 Redis	189
9.3.3 管道命令	191
9.3.4 密码登录	193
9.3.5 其他执行命令	194
9.4 与数据库交互的常见问题	194
9.4.1 连接池	194
9.4.2 读/写分离	197
9.4.3 分离配置文件和代码	197
9.5 小结	198
第 10 章 缓存利器	199
10.1 worker 进程的共享内存	200
10.1.1 创建共享内存区域	200
10.1.2 操作共享内存	201
10.1.3 制造消息队列	205
10.1.4 lua-resty-core	207
10.1.5 配置环境	208
10.2 Lua 模块下的共享内存	209
10.2.1 安装 lua-resty-lrucache	209
10.2.2 使用 lua-resty-lrucache 进行缓存的方法	209
10.3 当前请求在各执行阶段间的数据共享	213
10.3.1 ngx.ctx 的使用	213
10.3.2 子请求和内部重定向的缓存区别	214
10.4 利用共享内存配置动态 IP 地址认证	215
10.5 缓存和数据库的交互	218
10.5.1 从数据库获取数据	218
10.5.2 避免出现因缓存失效引起的“风暴”	223
10.6 小结	228
第 11 章 动态管理 upstream	229
11.1 实战需求分析	230

11.2	ngx_http_dyups_module	230
11.2.1	安装 ngx_http_dyups_module	230
11.2.2	动态管理 upstream	230
11.2.3	确保 upstream 数据的完整性	232
11.3	nginx-upsync-module	233
11.3.1	安装 nginx-upsync-module 和 Consul	233
11.3.2	Consul 的键值操作	234
11.3.3	动态管理 upstream	235
11.3.4	验证动态配置功能	237
11.3.5	高可用、高并发设计	237
11.4	基于 balancer_by_lua_block 的灵活控制	238
11.5	小结	239
第 12 章	Nginx 日志分析系统	240
12.1	实战需求分析	240
12.2	ngxtop 实时分析	241
12.3	Flume 方案的日志分析	243
12.4	智能化 nginx_log_analysis	244
12.4.1	架构重构	244
12.4.2	日志远程传输	245
12.4.3	时序数据库	245
12.4.4	日志规则设计	245
12.5	lua-resty-logger-socket 传输方案	246
12.5.1	安装 lua-resty-logger-socket	246
12.5.2	远程传输配置	247
12.5.3	参数解读	248
12.6	时序数据库 InfluxDB	249
12.6.1	安装 InfluxDB	249
12.6.2	基本概念和操作	249
12.6.3	数据分析之查询函数	250
12.6.4	数据存放之保留策略	251
12.6.5	定时任务之连续查询	251
12.6.6	客户端操作之 API	252

12.6.7 使用 UDP 模式传输数据	253
12.7 利用 lua-resty-http 实现 API 交互	254
12.7.1 安装 lua-resty-http	254
12.7.2 使用方式	254
12.8 提升 InfluxDB 性能	255
12.9 小结	255
第 13 章 静态容灾系统	256
13.1 荆棘之路	257
13.2 设计之路	259
13.3 架构流程图	261
13.3.1 反向代理系统	261
13.3.2 日志分析系统	261
13.3.3 后台系统	261
13.3.4 爬虫系统	262
13.3.5 容灾的缓存系统	262
13.3.6 时间版本的用途	263
13.3.7 异地容灾	263
13.4 核心代码解说	264
13.4.1 Ngx_Lua 应用	264
13.4.2 爬虫和日志系统的关系	266
13.4.3 全部容灾和部分容灾功能	266
13.5 静态容灾的智能关闭方案	267
13.5.1 从日志分析系统中复制请求	267
13.5.2 利用 goreplay 复制流量	267
13.5.3 Nginx 的镜像功能	268
13.5.4 灰度验证容灾系统缓存	269
13.6 小结	269
第 14 章 深入挖掘反向代理	270
14.1 验证码防御中心	270
14.2 鉴权管理中心	272
14.2.1 利用 auth_request 管理鉴权	272
14.2.2 利用 Ngx_Lua 子请求实现鉴权功能	273

14.3	并行访问	274
14.3.1	轻线程的启动和终止	275
14.3.2	等待和终止轻线程	276
14.3.3	URL 的外部合并和内部并发	278
14.3.4	使用 cosocket 实现外部访问	281
14.4	小结	281
第 15 章	爬虫	282
15.1	区分搜索引擎爬虫和恶意爬虫	282
15.2	应对搜索引擎爬虫	284
15.2.1	搜索引擎的 User-Agent	284
15.2.2	Robots 协议	285
15.2.3	控制搜索引擎爬虫实战	286
15.3	应对恶意爬虫	288
15.3.1	发现恶意爬虫	288
15.3.2	抵御恶意爬虫之禁止访问	289
15.3.3	抵御恶意爬虫之验证码拦截	290
15.4	小插曲——使用假数据迷惑恶意爬虫	290
15.5	小结	291
第 16 章	性能分析和优化	292
16.1	性能分析场景搭建	292
16.1.1	安装 SystemTap	292
16.1.2	LuaJIT 的 Debug 模式	293
16.1.3	开启 PCRE 的 Debug 模式	294
16.1.4	分析工具下载	294
16.1.5	找出不支持 Debug 模式的 lib 库	295
16.2	流量复制	295
16.3	各项指标分析和优化建议	295
16.3.1	连接池使用状态分析	295
16.3.2	找出读/写频繁的文件	297
16.3.3	执行阶段耗时分析	297
16.3.4	HTTP 连接数和文件打开数分析	298
16.3.5	找出 CPU “偷窃者”	298