

《深入理解Nginx（第2版）》

图书基本信息

书名：《深入理解Nginx（第2版）》

13位ISBN编号：9787111526252

出版时间：2016-2

作者：陶辉

页数：624

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：www.tushu000.com

《深入理解Nginx（第2版）》

内容概要

本书致力于说明开发Nginx模块的必备知识，第1版发行以后，深受广大读者的喜爱。然而由于Nginx功能繁多且性能强大，以致必须了解的基本技能也很庞杂，而第1版成书匆忙，缺失了几个进阶的技巧描述，因此第2版在此基础上进行了完善。

书中首先通过介绍官方Nginx的基本用法和配置规则，帮助读者了解一般Nginx模块的用法，然后重点介绍了如何开发HTTP模块(含HTTP过滤模块)来得到定制化的Nginx，其中包括开发一个功能复杂的模块所需要了解的各种知识，并对内存池的实现细节及TCP协议进行了详细介绍；接着，综合Nginx框架代码分析了Nginx架构的设计理念和技巧，此外，还新增了如何在模块中支持HTTP变量，以及与slab共享内存等相关的内容，相信通过完善，可进一步帮助读者更好地开发出功能丰富、性能一流的Nginx模块。

《深入理解Nginx（第2版）》

作者简介

陶辉，毕业于西安交通大学计算机科学与技术专业，曾就职于华为中央软件部、腾讯QQ空间、思科中国CRDC等公司，目前在阿里巴巴云计算公司的飞天团队工作，研究方向为介于IaaS和PaaS间的弹性计算，多年以来专注于Nginx的定制化应用，对Nginx的设计与特性有深刻认识，实战经验丰富，编写过许多优秀的Nginx模块并应用于企业级产品中，同时撰写了大量关于Nginx的技术文章。擅长Linux下高性能服务器的开发，以及分布式环境下海量数据存储的设计开发。

书籍目录

第一部分 Nginx能帮我们做什么

第1章 研究Nginx前的准备工作 2

- 1.1 Nginx是什么 2
- 1.2 为什么选择Nginx 5
- 1.3 准备工作 7
 - 1.3.1 Linux操作系统 7
 - 1.3.2 使用Nginx的必备软件 7
 - 1.3.3 磁盘目录 8
 - 1.3.4 Linux内核参数的优化 9
 - 1.3.5 获取Nginx源码 10
- 1.4 编译安装Nginx 11
- 1.5 configure详解 11
 - 1.5.1 configure的命令参数 11
 - 1.5.2 configure执行流程 18
 - 1.5.3 configure生成的文件 21
- 1.6 Nginx的命令行控制 23
- 1.7 小结 27

第2章 Nginx的配置 28

- 2.1 运行中的Nginx进程间的关系 28
- 2.2 Nginx配置的通用语法 31
 - 2.2.1 块配置项 31
 - 2.2.2 配置项的语法格式 32
 - 2.2.3 配置项的注释 33
 - 2.2.4 配置项的单位 33
 - 2.2.5 在配置中使用变量 33
- 2.3 Nginx服务的基本配置 34
 - 2.3.1 用于调试进程和定位问题的配置项 34
 - 2.3.2 正常运行的配置项 36
 - 2.3.3 优化性能的配置项 37
 - 2.3.4 事件类配置项 39
- 2.4 用HTTP核心模块配置一个静态Web服务器 40
 - 2.4.1 虚拟主机与请求的分发 41
 - 2.4.2 文件路径的定义 45
 - 2.4.3 内存及磁盘资源的分配 47
 - 2.4.4 网络连接的设置 49
 - 2.4.5 MIME类型的设置 52
 - 2.4.6 对客户端请求的限制 53
 - 2.4.7 文件操作的优化 54
 - 2.4.8 对客户端请求的特殊处理 56
 - 2.4.9 ngx_http_core_module模块提供的变量 57
- 2.5 用HTTP proxy module配置一个反向代理服务器 59
 - 2.5.1 负载均衡的基本配置 61
 - 2.5.2 反向代理的基本配置 63
- 2.6 小结 66

第二部分 如何编写HTTP模块

第3章 开发一个简单的HTTP模块 68

- 3.1 如何调用HTTP模块 68

- 3.2 准备工作 70
 - 3.2.1 整型的封装 71
 - 3.2.2 ngx_str_t数据结构 71
 - 3.2.3 ngx_list_t数据结构 71
 - 3.2.4 ngx_table_elt_t数据结构 75
 - 3.2.5 ngx_buf_t数据结构 75
 - 3.2.6 ngx_chain_t数据结构 77
- 3.3 如何将自己的HTTP模块编译进Nginx 77
 - 3.3.1 config文件的写法 77
 - 3.3.2 利用configure脚本将定制模块加入到Nginx中 78
 - 3.3.3 直接修改Makefile文件 81
- 3.4 HTTP模块的数据结构 82
- 3.5 定义自己的HTTP模块 86
- 3.6 处理用户请求 89
 - 3.6.1 处理方法的返回值 89
 - 3.6.2 获取URI和参数 92
 - 3.6.3 获取HTTP头部 94
 - 3.6.4 获取HTTP包体 97
- 3.7 发送响应 99
 - 3.7.1 发送HTTP头部 99
 - 3.7.2 将内存中的字符串作为包体发送 101
 - 3.7.3 经典的“Hello World”示例 102
- 3.8 将磁盘文件作为包体发送 103
 - 3.8.1 如何发送磁盘中的文件 104
 - 3.8.2 清理文件句柄 106
 - 3.8.3 支持用户多线程下载和断点续传 107
- 3.9 用C++语言编写HTTP模块 108
 - 3.9.1 编译方式的修改 108
 - 3.9.2 程序中的符号转换 109
- 3.10 小结 110
- 第4章 配置、error日志和请求上下文 111
 - 4.1 http配置项的使用场景 111
 - 4.2 怎样使用http配置 113
 - 4.2.1 分配用于保存配置参数的数据结构 113
 - 4.2.2 设定配置项的解析方式 115
 - 4.2.3 使用14种预设方法解析配置项 121
 - 4.2.4 自定义配置项处理方法 131
 - 4.2.5 合并配置项 133
 - 4.3 HTTP配置模型 135
 - 4.3.1 解析HTTP配置的流程 136
 - 4.3.2 HTTP配置模型的内存布局 139
 - 4.3.3 如何合并配置项 142
 - 4.3.4 预配置项处理方法的工作原理 144
 - 4.4 error日志的用法 145
 - 4.5 请求的上下文 149
 - 4.5.1 上下文与全异步Web服务器的关系 149
 - 4.5.2 如何使用HTTP上下文 151
 - 4.5.3 HTTP框架如何维护上下文结构 152
 - 4.6 小结 153

- 第5章 访问第三方服务 154
 - 5.1 upstream的使用方式 155
 - 5.1.1 ngx_http_upstream_t结构体 158
 - 5.1.2 设置upstream的限制性参数 159
 - 5.1.3 设置需要访问的第三方服务器地址 160
 - 5.1.4 设置回调方法 161
 - 5.1.5 如何启动upstream机制 161
 - 5.2 回调方法的执行场景 162
 - 5.2.1 create_request回调方法 162
 - 5.2.2 reinit_request回调方法 164
 - 5.2.3 finalize_request回调方法 165
 - 5.2.4 process_header回调方法 165
 - 5.2.5 rewrite_redirect回调方法 167
 - 5.2.6 input_filter_init与input_filter回调方法 167
 - 5.3 使用upstream的示例 168
 - 5.3.1 upstream的各种配置参数 168
 - 5.3.2 请求上下文 170
 - 5.3.3 在create_request方法中构造请求 170
 - 5.3.4 在process_header方法中解析包头 171
 - 5.3.5 在finalize_request方法中释放资源 175
 - 5.3.6 在ngx_http_mytest_handler方法中启动upstream 175
 - 5.4 subrequest的使用方式 177
 - 5.4.1 配置子请求的处理方式 177
 - 5.4.2 实现子请求处理完毕时的回调方法 178
 - 5.4.3 处理父请求被重新激活后的回调方法 179
 - 5.4.4 启动subrequest子请求 179
 - 5.5 subrequest执行过程中的主要场景 180
 - 5.5.1 如何启动subrequest 180
 - 5.5.2 如何转发多个子请求的响应包体 182
 - 5.5.3 子请求如何激活父请求 185
 - 5.6 subrequest使用的例子 187
 - 5.6.1 配置文件中子请求的设置 187
 - 5.6.2 请求上下文 188
 - 5.6.3 子请求结束时的处理方法 188
 - 5.6.4 父请求的回调方法 189
 - 5.6.5 启动subrequest 190
 - 5.7 小结 191
- 第6章 开发一个简单的HTTP过滤模块 192
 - 6.1 过滤模块的意义 192
 - 6.2 过滤模块的调用顺序 193
 - 6.2.1 过滤链表是如何构成的 194
 - 6.2.2 过滤链表的顺序 196
 - 6.2.3 官方默认HTTP过滤模块的功能简介 197
 - 6.3 HTTP过滤模块的开发步骤 198
 - 6.4 HTTP过滤模块的简单例子 200
 - 6.4.1 如何编写config文件 201
 - 6.4.2 配置项和上下文 201
 - 6.4.3 定义HTTP过滤模块 203
 - 6.4.4 初始化HTTP过滤模块 204

- 6.4.5 处理请求中的HTTP头部 204
- 6.4.6 处理请求中的HTTP包体 206
- 6.5 小结 206
- 第7章 Nginx提供的高级数据结构 207
 - 7.1 Nginx提供的高级数据结构概述 207
 - 7.2 ngx_queue_t双向链表 209
 - 7.2.1 为什么设计ngx_queue_t双向链表 209
 - 7.2.2 双向链表的使用方法 209
 - 7.2.3 使用双向链表排序的例子 212
 - 7.2.4 双向链表是如何实现的 213
 - 7.3 ngx_array_t动态数组 215
 - 7.3.1 为什么设计ngx_array_t动态数组 215
 - 7.3.2 动态数组的使用方法 215
 - 7.3.3 使用动态数组的例子 217
 - 7.3.4 动态数组的扩容方式 218
 - 7.4 ngx_list_t单向链表 219
 - 7.5 ngx_rbtree_t红黑树 219
 - 7.5.1 为什么设计ngx_rbtree_t红黑树 219
 - 7.5.2 红黑树的特性 220
 - 7.5.3 红黑树的使用方法 222
 - 7.5.4 使用红黑树的简单例子 225
 - 7.5.5 如何自定义添加成员方法 226
 - 7.6 ngx_radix_tree_t基数树 228
 - 7.6.1 ngx_radix_tree_t基数树的原理 228
 - 7.6.2 基数树的使用方法 230
 - 7.6.3 使用基数树的例子 231
 - 7.7 支持通配符的散列表 232
 - 7.7.1 ngx_hash_t基本散列表 232
 - 7.7.2 支持通配符的散列表 235
 - 7.7.3 带通配符散列表的使用例子 241
 - 7.8 小结 245
- 第三部分 深入Nginx
- 第8章 Nginx基础架构 248
 - 8.1 Web服务器设计中的关键约束 249
 - 8.2 Nginx的架构设计 251
 - 8.2.1 优秀的模块化设计 251
 - 8.2.2 事件驱动架构 254
 - 8.2.3 请求的多阶段异步处理 256
 - 8.2.4 管理进程、多工作进程设计 259
 - 8.2.5 平台无关的代码实现 259
 - 8.2.6 内存池的设计 259
 - 8.2.7 使用统一管道过滤器模式的HTTP过滤模块 260
 - 8.2.8 其他一些用户模块 260
 - 8.3 Nginx框架中的核心结构体ngx_cycle_t 260
 - 8.3.1 ngx_listening_t结构体 261
 - 8.3.2 ngx_cycle_t结构体 262
 - 8.3.3 ngx_cycle_t支持的方法 264
 - 8.4 Nginx启动时框架的处理流程 266
 - 8.5 worker进程是如何工作的 269

- 8.6 master进程是如何工作的 271
- 8.7 ngx_pool_t内存池 276
- 8.8 小结 284
- 第9章 事件模块 285
 - 9.1 事件处理框架概述 286
 - 9.2 Nginx事件的定义 288
 - 9.3 Nginx连接的定义 291
 - 9.3.1 被动连接 292
 - 9.3.2 主动连接 295
 - 9.3.3 ngx_connection_t连接池 296
 - 9.4 ngx_events_module核心模块 297
 - 9.4.1 如何管理所有事件模块的配置项 299
 - 9.4.2 管理事件模块 300
 - 9.5 ngx_event_core_module事件模块 302
 - 9.6 epoll事件驱动模块 308
 - 9.6.1 epoll的原理和用法 308
 - 9.6.2 如何使用epoll 310
 - 9.6.3 ngx_epoll_module模块的实现 312
 - 9.7 定时器事件 320
 - 9.7.1 缓存时间的管理 320
 - 9.7.2 缓存时间的精度 323
 - 9.7.3 定时器的实现 323
 - 9.8 事件驱动框架的处理流程 324
 - 9.8.1 如何建立新连接 325
 - 9.8.2 如何解决“惊群”问题 327
 - 9.8.3 如何实现负载均衡 329
 - 9.8.4 post事件队列 330
 - 9.8.5 ngx_process_events_and_timers流程 331
 - 9.9 文件的异步I/O 334
 - 9.9.1 Linux内核提供的文件异步I/O 335
 - 9.9.2 ngx_epoll_module模块中实现的针对文件的异步I/O 337
 - 9.10 TCP协议与Nginx 342
 - 9.11 小结 347
- 第10章 HTTP框架的初始化 348
 - 10.1 HTTP框架概述 349
 - 10.2 管理HTTP模块的配置项 352
 - 10.2.1 管理main级别下的配置项 353
 - 10.2.2 管理server级别下的配置项 355
 - 10.2.3 管理location级别下的配置项 358
 - 10.2.4 不同级别配置项的合并 364
 - 10.3 监听端口的管理 367
 - 10.4 server的快速检索 370
 - 10.5 location的快速检索 370
 - 10.6 HTTP请求的11个处理阶段 372
 - 10.6.1 HTTP处理阶段的普适规则 374
 - 10.6.2 NGX_HTTP_POST_READ_PHASE阶段 375
 - 10.6.3 NGX_HTTP_SERVER_REWRITE_PHASE阶段 378
 - 10.6.4 NGX_HTTP_FIND_CONFIG_PHASE阶段 378
 - 10.6.5 NGX_HTTP_REWRITE_PHASE阶段 378

- 10.6.6 NGX_HTTP_POST_REWRITE_PHASE阶段 379
- 10.6.7 NGX_HTTP_PREACCESS_PHASE阶段 379
- 10.6.8 NGX_HTTP_ACCESS_PHASE阶段 379
- 10.6.9 NGX_HTTP_POST_ACCESS_PHASE阶段 380
- 10.6.10 NGX_HTTP_TRY_FILES_PHASE阶段 380
- 10.6.11 NGX_HTTP_CONTENT_PHASE阶段 380
- 10.6.12 NGX_HTTP_LOG_PHASE阶段 382
- 10.7 HTTP框架的初始化流程 382
- 10.8 小结 384
- 第11章 HTTP框架的执行流程 385
 - 11.1 HTTP框架执行流程概述 386
 - 11.2 新连接建立时的行为 387
 - 11.3 第一次可读事件的处理 388
 - 11.4 接收HTTP请求行 394
 - 11.5 接收HTTP头部 398
 - 11.6 处理HTTP请求 400
 - 11.6.1 ngx_http_core_generic_phase 406
 - 11.6.2 ngx_http_core_rewrite_phase 408
 - 11.6.3 ngx_http_core_access_phase 409
 - 11.6.4 ngx_http_core_content_phase 412
 - 11.7 subrequest与post请求 415
 - 11.8 处理HTTP包体 417
 - 11.8.1 接收包体 419
 - 11.8.2 放弃接收包体 425
 - 11.9 发送HTTP响应 429
 - 11.9.1 ngx_http_send_header 430
 - 11.9.2 ngx_http_output_filter 432
 - 11.9.3 ngx_http_writer 435
 - 11.10 结束HTTP请求 437
 - 11.10.1 ngx_http_close_connection 438
 - 11.10.2 ngx_http_free_request 439
 - 11.10.3 ngx_http_close_request 440
 - 11.10.4 ngx_http_finalize_connection 441
 - 11.10.5 ngx_http_terminate_request 443
 - 11.10.6 ngx_http_finalize_request 443
 - 11.11 小结 446
- 第12章 upstream机制的设计与实现 447
 - 12.1 upstream机制概述 448
 - 12.1.1 设计目的 448
 - 12.1.2 ngx_http_upstream_t数据结构的意义 450
 - 12.1.3 ngx_http_upstream_conf_t配置结构体 453
 - 12.2 启动upstream 455
 - 12.3 与上游服务器建立连接 457
 - 12.4 发送请求到上游服务器 460
 - 12.5 接收上游服务器的响应头部 463
 - 12.5.1 应用层协议的两段划分方式 463
 - 12.5.2 处理包体的3种方式 464
 - 12.5.3 接收响应头部的流程 465
 - 12.6 不转发响应时的处理流程 469

- 12.6.1 input_filter方法的设计 469
- 12.6.2 默认的input_filter方法 470
- 12.6.3 接收包体的流程 472
- 12.7 以下游网速优先来转发响应 473
 - 12.7.1 转发响应的包头 474
 - 12.7.2 转发响应的包体 477
- 12.8 以上游网速优先来转发响应 481
 - 12.8.1 ngx_event_pipe_t结构体的意义 481
 - 12.8.2 转发响应的包头 485
 - 12.8.3 转发响应的包体 487
 - 12.8.4 ngx_event_pipe_read_upstream方法 489
 - 12.8.5 ngx_event_pipe_write_to_downstream方法 494
- 12.9 结束upstream请求 496
- 12.10 小结 499
- 第13章 邮件代理模块 500
 - 13.1 邮件代理服务器的功能 500
 - 13.2 邮件模块的处理框架 503
 - 13.2.1 一个请求的8个独立处理阶段 503
 - 13.2.2 邮件类模块的定义 504
 - 13.2.3 邮件框架的初始化 506
 - 13.3 初始化请求 506
 - 13.3.1 描述邮件请求的ngx_mail_session_t结构体 506
 - 13.3.2 初始化邮件请求的流程 509
 - 13.4 接收并解析客户端请求 509
 - 13.5 邮件认证 510
 - 13.5.1 ngx_mail_auth_http_ctx_t结构体 510
 - 13.5.2 与认证服务器建立连接 511
 - 13.5.3 发送请求到认证服务器 513
 - 13.5.4 接收并解析响应 514
 - 13.6 与上游邮件服务器间的认证交互 514
 - 13.6.1 ngx_mail_proxy_ctx_t结构体 516
 - 13.6.2 向上游邮件服务器发起连接 516
 - 13.6.3 与邮件服务器认证交互的过程 518
 - 13.7 透传上游邮件服务器与客户端间的流 520
 - 13.8 小结 524
- 第14章 进程间的通信机制 525
 - 14.1 概述 525
 - 14.2 共享内存 526
 - 14.3 原子操作 530
 - 14.3.1 不支持原子库下的原子操作 530
 - 14.3.2 x86架构下的原子操作 531
 - 14.3.3 自旋锁 533
 - 14.4 Nginx频道 535
 - 14.5 信号 538
 - 14.6 信号量 540
 - 14.7 文件锁 541
 - 14.8 互斥锁 544
 - 14.8.1 文件锁实现的ngx_shmtx_t锁 546
 - 14.8.2 原子变量实现的ngx_shmtx_t锁 548

- 14.9 小结 553
- 第15章 变量 554
 - 15.1 使用内部变量开发模块 555
 - 15.1.1 定义模块 556
 - 15.1.2 定义http模块加载方式 557
 - 15.1.3 解析配置中的变量 558
 - 15.1.4 处理请求 560
 - 15.2 内部变量工作原理 561
 - 15.2.1 何时定义变量 561
 - 15.2.2 相关数据结构详述 564
 - 15.2.3 定义变量的方法 572
 - 15.2.4 使用变量的方法 572
 - 15.2.5 如何解析变量 573
 - 15.3 定义内部变量 576
 - 15.4 外部变量与脚本引擎 577
 - 15.4.1 相关数据结构 578
 - 15.4.2 编译“set”脚本 581
 - 15.4.3 脚本执行流程 586
 - 15.5 小结 589
- 第16章 slab共享内存 590
 - 16.1 操作slab共享内存的方法 590
 - 16.2 使用slab共享内存池的例子 592
 - 16.2.1 共享内存中的数据结构 593
 - 16.2.2 操作共享内存中的红黑树与链表 595
 - 16.2.3 解析配置文件 600
 - 16.2.4 定义模块 603
 - 16.3 slab内存管理的实现原理 605
 - 16.3.1 内存结构布局 607
 - 16.3.2 分配内存流程 613
 - 16.3.3 释放内存流程 617
 - 16.3.4 如何使用位操作 619
 - 16.3.5 slab内存池间的管理 624
 - 16.4 小结 624

《深入理解Nginx（第2版）》

精彩短评

- 1、ng非常经典的c工程，看书后看代码快些
- 2、好书啊好书，除了讲解nginx，还深入详解了相关知识。作者表达能力超强。例如，我以前一直不太理解epoll，居然是从一本讲nginx的书里面，学会了epool.....。给作者点赞！
- 3、写 nginx 模块全靠它了，无论是当手册查还是了解 nginx 架构 都很好

《深入理解Nginx（第2版）》

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:www.tushu000.com