

系统归纳和深刻解读PHP开发中的编程思想、底层原理、核心技术、开发技巧、编码规范和最佳实践，为PHP程序员进阶修炼提供全面而高效的指导！



繁体版台湾发行



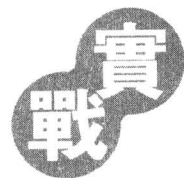
列旭松 陈文 著

PHP Core Technology and Best Practice

PHP 核心技术与最佳实践



机械工业出版社
China Machine Press



PHP Core Technology and Best Practice

PHP

核心技术与最佳实践

列旭松 陈文 著



机械工业出版社
China Machine Press

这是一本致力于为希望成为中高级 PHP 程序员的读者提供高效而有针对性指导的经典著作。本书系统归纳和深刻解读了 PHP 开发中的编程思想、底层原理、核心技术、开发技巧、编码规范和最佳实践。

全书分为 5 个部分：第一部分（1~2 章）从不同的角度阐述了面向对象软件设计思想的核心概念、技术和原则，分析了面向对象的特性、设计模式的理念，指出了如何设计低耦合、高可扩展性的软件，等等；第二部分（3~6 章）详细讲解了 PHP 中正则表达式的规范和使用技巧，PHP 网络编程的原理、方法、技巧和一些重要的操作，PDO、数据库应用优化，数据库设计和 MySQL 的高级应用，PHP 扩展引擎的原理与实践；第三部分（第 7 章）拨云见日，围绕 PHP 扩展开发进行了细致而深入的探讨，解析了 PHP 的底层实现和 Zend 虚拟机 API，并用 PHP 扩展开发的实例带领读者走进 PHP 的底层世界，旨在让读者对 PHP 性能优化、底层原理进行深入的理解。第四部分（8~11 章）重点讨论了缓存的设计、Memcached 的原理与实践、NoSQL 数据库 Redis 源码分析与应用实践、高性能 PHP 网站的架构和设计等内容；第五部分（12~14 章）详细讲解了 PHP 代码的调试和测试、Hash 算法和数据库的实现，以及 PHP 的编码规范，旨在帮助读者提高开发效率，养成良好编程习惯。

封底无防伪标均为盗版

版权所有，侵权必究

本书法律顾问 北京市展达律师事务所

图书在版编目（CIP）数据

PHP 核心技术与最佳实践 / 列旭松，陈文著. —北京：机械工业出版社，2012.12

ISBN 978-7-111-40192-6

I. P… II. ①列… ②陈… III. PHP 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字（2012）第 251198 号

机械工业出版社（北京市西城区百万庄大街 22 号 邮政编码 100037）

责任编辑：孙海亮

藁城市京瑞印刷有限公司印刷

2013 年 1 月第 1 版第 1 次印刷

186mm×240mm·33.75 印张

标准书号：ISBN 978-7-111-40192-6

定价：89.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

客服热线：(010) 88378991；88361066

购书热线：(010) 68326294；88379649；68995259

投稿热线：(010) 88379604

读者信箱：hzjsj@hzbook.com

前 言

为什么要写这本书

近几年，市场上关于 PHP 的书已经很多了，各种培训机构也如雨后春笋般不断增加。那为什么还要写这本书呢？这本书存在的意义又在哪里？这要从下面的几个问题说起。

有没有这样一本 PHP 教材，它不讲 HTML 和 CSS，也不讲 JavaScript 基础，甚至不讲 PHP 语法基础？

有没有这样一本 PHP 教材，它不讲留言板或博客的开发，也不讲数据库的 CRUD 操作？

有没有这样一本 PHP 教材，它专注于 Web 开发技术的最前沿，深入浅出，适合中高级程序员的进阶和提高？

有没有这样一本 PHP 教材，它提倡面向对象的程序思想，提倡算法和数据结构的重要性，提倡对网络协议的深入理解，且没有大篇幅的代码，而是更多偏重于理论讲解？

有没有这样一本 PHP 教材，它探讨 PHP 的扩展开发，探讨高并发大流量的架构，深入探讨 NoSQL 的内部实现和细节？

以上几个问题也是我在早期 PHP 学习的过程中一直在寻找的答案，可是我并没有找到一本理想的 PHP 书籍，一本适合中高级程序员进阶的书籍。当怀着同样问题的旭松兄找到我时，我们不禁产生一个念头：“既然现在市场上缺少一本这样的书籍，我们何不自己写一本呢？利己利人的事值得去做。”然后一拍即合，说做就做，现在这本书经历长达一年多的酝酿和写作过程终于完稿了。

我是在大学期间接触到 PHP 语言的，并马上被其简洁的语法和极高的开发效率所吸引，一头扎进 PHP 开发的世界中。随着学习的深入，并经常关注 PHP 社区的动态，我很快意识到一些 PHP 社区普遍存在的问题。比如 PHP 社区一直争论算法重不重要，面向对象好不好，代码质量重要还是开发速度重要的问题。还有譬如为什么我去大型互联网公司应聘 PHP 程序员，却不考察我对 PHP 语法和函数的掌握情况，而是会问我 C 语言、算法、网络协议、高并发处理、MVC 理论这些看似和 PHP 不沾边的问题。

PHP 到底要怎么学，学什么，一个高级 PHP 程序员应该是什么样的，我想这也是很多 PHP 新手和工作一两年的 PHP 开发者的疑惑。这本书所要解决的就是这一问题系列的问题。

在我看来，一本技术书籍的价值在于其对知识的提炼和与众不同的地方。举例来说，到一个书店去看书，你最想用笔抄下来或撕下来带走的那几页，就是对你帮助最大的东西，也是你

认为这本书的价值所在。也是基于这个想法，我们思考这本书该写什么，怎么写，哪些地方对读者有帮助。我们试图从不同的角度带领读者来看 PHP，进而给这本书注入一些不一样的东西。我们希望这是一件有意义的事。

本书适合的对象

- ❑ PHP 爱好者；
- ❑ 想进阶的初级 PHP 程序员；
- ❑ 对 PHP 扩展开发感兴趣的读者；
- ❑ 对高并发感兴趣的读者；
- ❑ 对 NoSQL 应用和实现原理感兴趣的读者；
- ❑ 从事 PHP 网络应用，想知道 HTTP 协议、Socket 等更多细节的开发人员；
- ❑ 想就职于大型互联网公司的 PHP 程序员；
- ❑ 开设相关课程的大专院校的学生；
- ❑ 公司内部培训的学员。

如何阅读本书

本书一共有 14 章。每章节都可以单独阅读，由于部分知识点之间存在一定的衔接，故建议按先后顺序阅读。

第 1 章为面向对象思想的核心概念。本章主要讲解面向对象开发的思想，重点讲解面向对象模型的建立，以及面向对象的一些基础概念。通过大量对比和实例，尤其是与 Java 的对比，力图从不同角度讲解 PHP 面向对象的特性，让 PHP 程序员看到不同的面向对象。求同存异是本章的核心思想。

第 2 章为用面向对象思维写程序。本章用简练的语言讲解了面向对象设计的五大原则，这五大原则也是理解设计模式的基础所在，帮助读者站在一个更高的角度思考面向对象。

第 3 章为正则表达式技巧与实战。本章详细介绍了正则的基础语法，通过大量的示例、通俗的语言讲解正则概念，引导读者理解正则的一系列规则。接下来，结合实际工作安全过滤、URL 重写等实例，加深对正则的应用和掌握。最后给出正则效率优化的一些普遍技巧和替代方案，让读者对正则的使用得心应手。

第 4 章为 PHP 网络技术及应用。本章着重介绍了 HTTP 协议、Socket 开发、WebService、Cookie 和 Session 使用等。结合实战向读者阐述网络开发的核心和重点，特别是对 HTTP 协议的理解。HTTP 协议是 Web 开发的基石，也是各种面试和开发中必然遇到的知识点。而 Socket 则是应用交互的桥梁，保证了有用的可扩展性。

第 5 章为 PHP 与数据库基础。本章从不同角度分析了 MySQL，介绍了 PDO、MySQL 优化、

存储过程、事件调度机制以及 MySQL 安全防范等内容。

第6章为 PHP 模板引擎的原理与最佳实践。本章通过实现一个简单的模板引擎，学习模板引擎的原理和使用方法，然后对比几大流行的模板引擎实现方案，简单介绍了各种实现方案思想和优缺点，最后探讨模板引擎的意义。

第7章为 PHP 扩展开发。本章的知识是本书核心内容，介绍了 PHP 扩展开发的几个重要知识点，如扩展框架搭建、PHP 生命周期、PHP 变量在内核中的实现方式、Zend 引擎、内存管理等，让读者深入 PHP 底层，知其然也知其所以然。

第8章为缓存。本章主要介绍了缓存的基本原理和三个衡量指标，通过几个实例加深读者对缓存的理解。利用本章知识，读者应该能设计一个比较合理的缓存方案。

第9章为 Memcached 应用与内幕。本章深入剖析了 Memcached 的实现和内部结构，从而使读者掌握 Memcached 的高级应用，对构建复杂环境的缓存层有个清晰的认识。

第10章为 Redis 应用与内幕。本章重点介绍了 Redis 的深入应用，如事务处理、主从同步、虚拟内存等，和第9章类似，探讨了 Redis 的实现内幕。合理利用 Redis 可以为我们解决大流量高并发的应用。

第11章为高性能网站架构。本章探讨了高性能架构的基本出发点，重点以 HandlerSocket、MySQL 主从复制、反向代理缓存软件 Varnish 和任务分发框架 Gearman 为例，讲述几种高性能架构中会用到的技术。

第12章为调试与测试。科学的调试方法有助于快速找出潜在的 Bug、理解复杂应用的流程、提高开发效率。单元测试是代码质量的保障。在这一章的最后一节介绍了使用 JMeter 进行压力测试的方法。

第13章为 Hash 算法与数据库的实现。本章介绍了 Hash 算法的基本原理，用此算法实现一个简单的、基于 Hash 的数据库，让读者意识到算法的重要性和可操作性。

第14章为 PHP 编码规范。本章介绍了 PHP 开发中应遵循的基本代码规范，并提出合理建议。好的代码必然是规范的代码。

本书第1、2、3、5、6、8、12、14章由陈文撰写，第7、9、10、11、13章由列旭松撰写，第4章由两人共同完成。

勘误和支持

由于我们的水平和开发经验有限，同时计算机技术更新较快，书中难免存在不足之处，有些章节内容可能从未来的某一天开始不再适用，还望读者理解和体谅，并恳请读者批评指正。您若对本书有什么好的建议或者对书中部分内容有疑惑，可与我们联系，我们将尽量为读者提供最满意的解答。期待得到您的真挚反馈。我们的联系方式如下：

陈文：waitfox@qq.com

列旭松：liexusong@qq.com

感谢

首先要感谢 PHP 之父 Rasmus Lerdorf, 是他创建了这个简单、轻松、有趣、快速而又高效的语言; 其次, 感谢 PHP 社区每一位充满活力的朋友, 和你们的交流使我学到很多, 本书有不少内容就来自于社区的智慧。

在这里尤其要感谢机械工业出版社华章公司的大力支持, 特别是杨福川和白宇两位编辑, 在一年多的时间里, 因为有了你们的耐心指导、逐字逐句认真审稿和改稿才有了本书的诞生。

最后, 还要感谢家人和朋友的支持。

陈文

林文和吴博

陈文: waitong@pp.com

吴博: liuzhong@pp.com

目 录

前言

第 1 章 面向对象思想的核心概念 / 1

- 1.1 面向对象的“形”与“本” / 1
 - 1.1.1 对象的“形” / 2
 - 1.1.2 对象的“本” / 4
 - 1.1.3 对象与数组 / 5
 - 1.1.4 对象与类 / 5
- 1.2 魔术方法的应用 / 7
 - 1.2.1 __set 和 __get 方法 / 8
 - 1.2.2 __call 和 __callStatic 方法 / 9
 - 1.2.3 __toString 方法 / 11
- 1.3 继承与多态 / 14
 - 1.3.1 类的组合与继承 / 14
 - 1.3.2 各种语言中的多态 / 19
- 1.4 面向接口编程 / 23
 - 1.4.1 接口的作用 / 23
 - 1.4.2 对 PHP 接口的思考 / 25
- 1.5 反射 / 28
 - 1.5.1 如何使用反射 API / 28
 - 1.5.2 反射有什么作用 / 30
- 1.6 异常和错误处理 / 32
 - 1.6.1 如何使用异常处理机制 / 32
 - 1.6.2 怎样看 PHP 的异常 / 38
 - 1.6.3 PHP 中的错误级别 / 38
 - 1.6.4 PHP 中的错误处理机制 / 40
- 1.7 本章小结 / 42

第2章 面向对象的设计原则 / 44

- 2.1 面向对象设计的五大原则 / 44
 - 2.1.1 单一职责原则 / 44
 - 2.1.2 接口隔离原则 / 52
 - 2.1.3 开放-封闭原则 / 55
 - 2.1.4 替换原则 / 57
 - 2.1.5 依赖倒置原则 / 59
- 2.2 一个面向对象留言本的实例 / 62
- 2.3 面向对象的思考 / 65
- 2.4 本章小结 / 67

第3章 正则表达式基础与应用 / 68

- 3.1 认识正则表达式 / 68
 - 3.1.1 PHP 中的正则函数 / 68
 - 3.1.2 正则表达式的组成 / 69
 - 3.1.3 测试工具的使用 / 69
- 3.2 正则表达式中的元字符 / 70
 - 3.2.1 什么是元字符 / 71
 - 3.2.2 起始和结束元字符 / 72
 - 3.2.3 点号 / 73
 - 3.2.4 量词 / 74
- 3.3 正则表达式匹配规则 / 74
 - 3.3.1 字符组 / 74
 - 3.3.2 转义 / 75
 - 3.3.3 反义 / 76
 - 3.3.4 分支 / 77
 - 3.3.5 分组 / 78
 - 3.3.6 反向引用 / 79
 - 3.3.7 环视 / 81
 - 3.3.8 贪婪/懒惰匹配模式 / 83
- 3.4 构造正则表达式 / 84
 - 3.4.1 正则表达式的逻辑关系 / 85
 - 3.4.2 运算符优先级 / 87
 - 3.4.3 正则表达式的常用模式 / 88
- 3.5 正则在实际开发中的应用 / 92

- 3.5.1 移动手机校验 / 92
- 3.5.2 匹配 E-mail 地址 / 93
- 3.5.3 转义在数据安全中的应用 / 93
- 3.5.4 URL 重写与搜索引擎优化 / 95
- 3.5.5 删除文件中的空行和注释 / 98
- 3.6 正则表达式的效率与优化 / 99
- 3.7 本章小结 / 103

第4章 PHP 网络技术及应用 / 105

- 4.1 HTTP 协议详解 / 105
 - 4.1.1 HTTP 协议与 SPDY 协议 / 105
 - 4.1.2 HTTP 协议如何工作 / 106
 - 4.1.3 HTTP 应用：模拟灌水机器人 / 110
 - 4.1.4 垃圾信息防御措施 / 116
- 4.2 抓包工具 / 119
 - 4.2.1 抓包工具分类 / 119
 - 4.2.2 Fiddler 功能与原理 / 119
 - 4.2.3 安装 Fiddler / 120
 - 4.2.4 Fiddler 基本界面 / 121
 - 4.2.5 使用 Fiddler 进行 HTTP 断点调试 / 124
- 4.3 Socket 进程通信机制及应用 / 127
 - 4.3.1 进程通信相关概念 / 127
 - 4.3.2 Socket 演示：实现服务器端与客户端的交互 / 128
 - 4.3.3 Socket 函数原型 / 131
 - 4.3.4 PHP 中的 Socket 函数 / 131
 - 4.3.5 Socket 交互应用：使用 Socket 抓取数据 / 134
- 4.4 cURL 工具及应用 / 136
 - 4.4.1 建立 cURL 请求的基本步骤 / 136
 - 4.4.2 检查 cURL 错误和获取返回信息 / 137
 - 4.4.3 在 cURL 中伪造头信息 / 138
 - 4.4.4 在 cURL 中用 POST 方法发送数据 / 140
 - 4.4.5 使用 cURL 上传文件 / 141
 - 4.4.6 cURL 批处理 / 142
 - 4.4.7 cURL 设置项 / 144
 - 4.4.8 网络应用：使用 cURL 抓取腾讯微博 / 145
- 4.5 简单邮件传输协议 SMTP / 151

- 4.5.1 SMTP 协议如何工作 / 151
- 4.5.2 SMTP 协议常用命令 / 153
- 4.5.3 SMTP 协议应用：使用 Socket 发送邮件 / 153
- 4.6 Webservice 的前世今生 / 157
 - 4.6.1 Webservice 简介 / 157
 - 4.6.2 认识 PHPRPC 协议 / 159
 - 4.6.3 Web 服务的实现模式 / 165
 - 4.6.4 简单对象访问协议 SOAP / 166
 - 4.6.5 调试工具 soapUI / 168
- 4.7 Cookie 详解 / 170
 - 4.7.1 Cookie 的基本概念及设置 / 170
 - 4.7.2 PHP 和 JavaScript 对 Cookie 的操作 / 172
 - 4.7.3 Cookie 存储机制及应用 / 174
 - 4.7.4 Cookie 跨域与 P3P 协议 / 175
 - 4.7.5 本地存储 localStorage / 177
- 4.8 Session 详解 / 179
 - 4.8.1 Session 的基本概念及设置 / 179
 - 4.8.2 Session 的工作原理 / 180
 - 4.8.3 Session 入库 / 181
 - 4.8.4 Cookie 与 Session 问答 / 185
- 4.9 本章小结 / 186

第5章 PHP 与数据库基础 / 187

- 5.1 什么是 PDO / 187
 - 5.1.1 PDO 预定义类 / 188
 - 5.1.2 如何使用 PDO / 189
 - 5.1.3 PDO 参数绑定与预编译 / 189
 - 5.1.4 PDO 事务处理 / 191
 - 5.1.5 PDO 的效率问题 / 192
- 5.2 数据库应用优化 / 193
 - 5.2.1 基本语句优化 10 个原则 / 193
 - 5.2.2 索引与性能分析 / 194
 - 5.2.3 服务器和配置的优化 / 197
 - 5.2.4 MySQL 瓶颈及应对措施 / 199
- 5.3 数据库设计 / 200
 - 5.3.1 范式与反范式 / 200

- 5.3.2 数据库分区 / 201
- 5.3.3 分表的应用 / 203
- 5.4 MySQL 的高级应用 / 205
 - 5.4.1 MySQL 自增长序列 / 205
 - 5.4.2 MySQL 视图 / 207
 - 5.4.3 MySQL 存储过程和事件调度 / 210
 - 5.4.4 用 MySQL 模拟消息队列 / 212
 - 5.4.5 SQL 注入漏洞与防范 / 216
- 5.5 本章小结 / 220

第 6 章 PHP 模板引擎的原理与实践 / 221

- 6.1 代码分层的思想 / 221
- 6.2 实现一个简单的模板引擎骨架 / 224
 - 6.2.1 搭建模板引擎基础类骨架 / 224
 - 6.2.2 编译类骨架 / 226
 - 6.2.3 测试模板引擎 / 229
- 6.3 模板引擎的编译 / 230
 - 6.3.1 实现变量标签 / 230
 - 6.3.2 实现 foreach 标签 / 231
 - 6.3.3 实现 if...else 标签 / 233
 - 6.3.4 对 PHP 原生语法的支持 / 234
- 6.4 完善模板引擎 / 235
 - 6.4.1 模板缓存机制的实现 / 235
 - 6.4.2 调试和缓存清理 / 236
 - 6.4.3 如何使用模板 / 244
- 6.5 常用模板引擎 / 245
 - 6.5.1 Discuz 模板引擎 / 245
 - 6.5.2 Smarty 模板引擎 / 246
 - 6.5.3 DedeCms 模板引擎 / 247
 - 6.5.4 Blitz 模板引擎 / 248
 - 6.5.5 模板引擎的一些思考 / 249
- 6.6 本章小结 / 251

第 7 章 PHP 扩展开发 / 252

- 7.1 为什么要开发 PHP 扩展 / 252
- 7.2 搭建 PHP 扩展框架 / 252

- 7.2.1 PHP 源代码目录 / 252
- 7.2.2 ext_skel 工具 / 253
- 7.2.3 Windows 平台环境配置 / 253
- 7.2.4 Linux 平台环境配置 / 256
- 7.2.5 PHP 的生命周期 / 257
- 7.3 PHP 内核中的变量 / 260
 - 7.3.1 PHP 变量在内核中的存储方式 / 260
 - 7.3.2 PHP 内核变量访问宏 / 262
 - 7.3.3 引用计数器与写时复制 / 263
- 7.4 PHP 内核中的 HashTable 分析 / 265
 - 7.4.1 PHP 内核 HashTable 的数据结构 / 265
 - 7.4.2 HashTable 的代码实现 / 270
- 7.5 Zend API 详解与扩展编写 / 274
 - 7.5.1 什么是 Zend 引擎 / 274
 - 7.5.2 Zend 引擎内存管理 / 275
 - 7.5.3 PHP 扩展的架构 / 276
 - 7.5.4 接收用户传递的参数 / 283
 - 7.5.5 在 PHP 扩展中创建变量 / 290
 - 7.5.6 在 PHP 扩展中为变量赋值 / 293
 - 7.5.7 错误和输出 API / 304
 - 7.5.8 运行时信息函数 / 306
 - 7.5.9 调用用户自定义函数 / 307
 - 7.5.10 PHP 配置项 / 309
 - 7.5.11 创建常量的宏 / 312
- 7.6 编写一个完整的扩展 / 313
 - 7.6.1 链表结构的实现 / 313
 - 7.6.2 创建 PHP 扩展框架 / 319
 - 7.6.3 编写代码 / 319
 - 7.6.4 编译安装扩展 / 327
 - 7.6.5 测试扩展 / 328
- 7.7 本章小结 / 329

第 8 章 缓存详解 / 330

- 8.1 认识缓存 / 330
 - 8.1.1 为什么使用缓存 / 330
 - 8.1.2 命中率 / 331

- 8.1.3 缓存更新策略 / 332
- 8.1.4 缓存最大数据量 / 333
- 8.2 文件缓存 / 333
 - 8.2.1 文件缓存机制 / 333
 - 8.2.2 文件缓存开源产品 Secache / 335
- 8.3 Opcode 缓存 / 337
 - 8.3.1 eAccelerator 下载及使用 / 337
 - 8.3.2 如何查看 Opcode / 338
- 8.4 客户端缓存 / 340
 - 8.4.1 客户端缓存规则 / 340
 - 8.4.2 HTTP 协议中的缓存使用 / 341
 - 8.4.3 HTTP 缓存实例 / 343
 - 8.4.4 HTML 5 中的 Application Cache / 345
- 8.5 Web 服务器缓存 / 347
 - 8.5.1 Apache 缓存 / 347
 - 8.5.2 Nginx 缓存 / 348
- 8.6 本章小结 / 349

第 9 章 Memcached 使用与实践 / 350

- 9.1 为什么要用 Memcached / 350
- 9.2 Memcached 的安装及使用 / 350
 - 9.2.1 安装 Memcached 服务器 / 351
 - 9.2.2 安装 Memcached 客户端 / 352
 - 9.2.3 使用 memcache 扩展访问 Memcached 服务器 / 353
 - 9.2.4 使用 Memcached 加速 Web 应用 / 356
- 9.3 深入了解 Memcached / 357
 - 9.3.1 Memcached 如何支持高并发 / 357
 - 9.3.2 使用 Slab 分配算法保存数据 / 357
 - 9.3.3 删除过期 item / 361
 - 9.3.4 使用 LRU 算法淘汰数据 / 362
 - 9.3.5 Memcached 多线程模型 / 363
- 9.4 Memcached 分布式布置方案 / 366
 - 9.4.1 普通 Hash 分布 / 366
 - 9.4.2 一致性 Hash 分布 / 367
 - 9.4.3 一致性 Hash 分布算法实例 / 369
- 9.5 本章小结 / 371

第 10 章 Redis 使用与实践 / 372

10.1 Redis 的安装及使用 / 372

10.1.1 Redis 安装步骤 / 372

10.1.2 修改 Redis 配置文件 / 373

10.1.3 运行 Redis 服务器 / 374

10.1.4 key 相关命令 / 374

10.1.5 Redis 支持的数据类型 / 375

10.1.6 Redis 排序命令详解 / 382

10.2 事务处理 / 384

10.2.1 事务处理原理 / 384

10.2.2 事务处理实现 / 385

10.3 持久化 / 387

10.3.1 内存快照 / 387

10.3.2 日志追加 / 387

10.4 主从同步 / 388

10.4.1 Redis 主从同步原理 / 388

10.4.2 Slave 端的工作流程 / 389

10.4.3 Master 端的工作流程 / 392

10.5 虚拟内存 / 397

10.5.1 配置文件信息 / 397

10.5.2 开启 VM 的后台操作 / 398

10.5.3 Redis Object 和 VM Pointer / 399

10.5.4 交换过程 / 400

10.5.5 阻塞式 VM / 403

10.5.6 非阻塞式 VM / 405

10.6 扩展库 phpredis 安装及使用 / 408

10.7 Redis 应用实践 / 409

10.7.1 使用消息队列发布微博 / 409

10.7.2 Redis 替代文件存储 Session / 410

10.8 深入了解 Redis 内核 / 413

10.8.1 内存淘汰 / 413

10.8.2 对象引用计数器 / 415

10.8.3 自动关闭超时连接 / 418

10.8.4 清除过期数据 / 419

10.9 本章小结 / 421

第 11 章 高性能网站架构方案 / 422

- 11.1 如何优化网站响应时间 / 422
 - 11.1.1 吞吐率 / 423
 - 11.1.2 压力测试 / 423
 - 11.1.3 持久连接 / 426
- 11.2 MySQL 响应速度提高方案: HandlerSocket / 429
 - 11.2.1 HandlerSocket 工作原理 / 430
 - 11.2.2 HandlerSocket 安装和配置 / 430
 - 11.2.3 PHP-HandlerSocket 性能测试 / 432
- 11.3 MySQL 稳定性提高方案: 主从复制 / 433
 - 11.3.1 主从复制工作原理 / 433
 - 11.3.2 主从复制配置 / 434
 - 11.3.3 连接主从服务器 / 436
- 11.4 Web 应用加速方案: Varnish / 438
 - 11.4.1 传统代理与反向代理 / 438
 - 11.4.2 Varnish 安装和配置 / 439
 - 11.4.3 Varnish 性能测试 / 440
 - 11.4.4 修改缓存规则 / 442
 - 11.4.5 监控 Varnish 运行状态 / 444
- 11.5 异步计算方案: Gearman / 445
 - 11.5.1 Gearman 工作原理 / 445
 - 11.5.2 安装 Gearman 和 PHP 扩展 / 447
 - 11.5.3 使用 Gearman 异步发送邮件 / 447
- 11.6 本章小结 / 449

第 12 章 代码调试和测试 / 450

- 12.1 调试 PHP 代码 / 450
 - 12.1.1 PHP 调试函数 / 450
 - 12.1.2 断点调试与变量跟踪工具 Xdebug / 454
- 12.2 前端调试 / 457
 - 12.2.1 Firebug 调试 API / 457
 - 12.2.2 使用 Firebug 调试 DOM 结构 / 459
 - 12.2.3 使用 Firebug 调试 JavaScript / 460
 - 12.2.4 使用 Fiddler 调试远程服务器上的文件 / 462
- 12.3 日志管理 / 463

- 12.3.1 PHP 日志 / 463
- 12.3.2 Apache 服务器日志 / 464
- 12.3.3 MySQL 日志 / 465
- 12.4 代码性能测试技术 / 468
 - 12.4.1 时间点测试 / 468
 - 12.4.2 文件查看工具 WinCacheGrind / 469
 - 12.4.3 性能测试注意事项 / 471
- 12.5 单元测试 / 472
 - 12.5.1 单元测试框架 PHPUnit 的安装 / 472
 - 12.5.2 结合 NetBeans 使用 PHPUnit 进行单元测试 / 473
 - 12.5.3 PHPUnit 中的断言函数 / 476
 - 12.5.4 PHPUnit 常用方法 / 478
 - 12.5.5 PHPUnit 常用注解 / 478
- 12.6 压力测试 / 480
 - 12.6.1 使用 JMeter 压力测试 HTTP / 480
 - 12.6.2 压力测试 MySQL / 484
 - 12.6.3 JMeter + Badboy 组合测试 / 487
- 12.7 本章小结 / 489

第 13 章 Hash 算法与数据库实现 / 490

- 13.1 Hash 函数 / 490
- 13.2 Hash 算法 / 490
 - 13.2.1 直接取余法 / 490
 - 13.2.2 乘积取整法 / 490
 - 13.2.3 经典 Hash 算法 Times33 / 491
- 13.3 Hash 表 / 491
 - 13.3.1 Hash 表结构 / 491
 - 13.3.2 使用 PHP 实现 Hash 表 / 492
 - 13.3.3 Hash 表冲突 / 493
 - 13.3.4 拉链法解决冲突 / 494
- 13.4 一个小型数据库的实现 / 496
 - 13.4.1 pack 函数的用法 / 497
 - 13.4.2 unpack 函数的用法 / 499
 - 13.4.3 索引文件和数据文件 / 500
 - 13.4.4 数据库接口方法 / 501
 - 13.4.5 源代码解析 / 502