

高性能MySQL：第3版

作者：施瓦茨 (Baron Schwartz)

内容简介

本书是MySQL领域的经典之作，拥有广泛的影响力。第3版更新了大量的内容，不但涵盖了最新MySQL 5.5版本的新特性，也讲述了关于固态硬盘、高可扩展性设计和云计算环境下的数据库相关的新内容，原有的基准测试和性能优化部分也做了大量的扩展和补充。全书共分为16章和6个附录，内容涵盖MySQL架构和历史，基准测试和性能剖析，数据库软硬件性能优化，复制、备份和恢复，高可用与高可扩展性，以及云端的MySQL和MySQL相关工具等方面的内容。每一章都是相对独立的主题，读者可以有选择性地单独阅读。

本书不但适合数据库管理员（DBA）阅读，也适合开发人员参考学习。不管是数据库新手还是专家，相信都能从本书有所收获。

©2012 by Baron Schwartz, Peter Zaitsev, Vadim Tkachenko.

Simplified Chinese Edition, jointly published by O'Reilly Media, Inc. and Publishing House of Electronics Industry, 2013. Authorized translation of the English edition, 2012 O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

本书简体中文版专有出版权由O'Reilly Media, Inc.授予电子工业出版社。未经许可,不得以任何方式复制或抄袭本书的任何部分。专有出版权受法律保护。

版权贸易合同登记号图字: 01-2013-1661

图书在版编目(CIP)数据

高性能MySQL: 第3版 / (美)施瓦茨(Schwartz, B.), (美)扎伊采夫(Zaitsev, P.), (美)特卡琴科(Tkachenko, V.)著; 宁海元等译. —北京: 电子工业出版社, 2013.5

书名原文: High Performance MySQL, Third Edition

ISBN 978-7-121-19885-4

I. ①高... II. ①施... ②扎... ③特... ④宁... III. ①关系数据库系统 IV. ①TP311.138

中国版本图书馆CIP数据核字(2013)第054420号

策划编辑: 张春雨

责任编辑: 白涛 贾莉

封面设计: Karen Montgomery 张健

印刷: 三河市鑫金马印装有限公司

装订: 三河市鑫金马印装有限公司

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: 50

字数: 1040千字

印次: 2013年5月第1次印刷

定价: 128.00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。

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

服务热线: (010) 88258888。

O'Reilly Media, Inc. 介绍

O'Reilly Media通过图书、杂志、在线服务、调查研究和会议等方式传播创新知识。自1978年开始，O'Reilly一直都是前沿发展的见证者和推动者。超级极客们正在开创着未来，而我们关注真正重要的技术趋势——通过放大那些“细微的信号”来刺激社会对新科技的应用。作为技术社区中活跃的参与者，O'Reilly的发展充满了对创新的倡导、创造和发扬光大。

O'Reilly为软件开发人员带来革命性的“动物书”；创建第一个商业网站（GNN）；组织了影响深远的开放源代码峰会，以至于开源软件运动以此命名；创立了Make杂志，从而成为DIY革命的主要先锋；公司一如既往地通过多种形式缔结信息与人的纽带。O'Reilly的会议和峰会集聚了众多超级极客和高瞻远瞩的商业领袖，共同描绘出开创新产业的革命性思想。作为技术人士获取信息的选择，O'Reilly现在还将先锋专家的知识传递给普通的计算机用户。无论是通过书籍出版、在线服务或者面授课程，每一项O'Reilly的产品都反映了公司不可动摇的理念——信息是激发创新的力量。

业界评论

“O'Reilly Radar博客有口皆碑。”

——Wired

“O'Reilly凭借一系列（真希望当初我也想到了）非凡想法建立了数百万美元的业务。”

——Business 2.0

“O'Reilly Conference是聚集关键思想领袖的绝对典范。”

——CRN

“一本O'Reilly的书就代表一个有用、有前途、需要学习的主题。”

——Irish Times

“Tim是位特立独行的商人，他不光放眼于最长远、最广阔的视野并且切实地按照Yogi Berra的建议去做了：‘如果你在路上遇到岔路口，走小路（岔路）。’回顾过去Tim似乎每一次都选择了小路，而且有几次都是一闪即逝的机会，尽管大路也不错。”

——Linux Journal

译者序

在互联网行业，MySQL数据库毫无疑问已经是最常用的数据库。LAMP（Linux+Apache+MySQL+PHP）甚至已经成为专有名词，也是很多中小网站建站的首选技术架构。我所在的公司淘宝网，在2003年非典肆虐期间创立时，选择的的就是LAMP架构，当时MySQL的版本还是4.0。但是到了2003年底，由于业务超预期的增长，MySQL 4.0（当时用的还是MyISAM引擎）的很多缺点在高并发大压力下暴露了出来，于是技术上开始改用商业的Oracle数据库。随后几年Oracle加小型机和高端存储的数据库架构支撑了淘宝网业务的爆炸式增长，数据库也从最初的两三个库增长到十几个库，并且每个库的硬件已经逐步升级到顶配，“天花板”很明显地摆在了眼前。于是在2008年，基于PC服务器的MySQL数据库再次成为DBA团队的选择，这时候MySQL的稳定版本已经升级到5.0，并且5.1也已经在开发中，性能和特性相对于2003年的时候已经有了非常大的提升。淘宝网的数据库架构也逐渐从垂直拆分走向水平拆分，在大规模水平集群的架构设计中，开源的MySQL受到的关注度越来越高，并且一年多来的实践也证明了MySQL（存储引擎主要使用的是InnoDB）在高压下的可用性。于是从2009年开始，后来颇受外界关注的所谓“去IOE”开始实施，经过三年多的架构改造，到2012年整个淘宝网的核心交易系统已经全部运行在基于PC服务器的MySQL数据库集群中，全部实例数超过2000个。今年的“双11”大促中，MySQL单库经受了最高达6.5万的QPS，某个拥有32个节点的核心集群的总QPS则稳定在86万以上，并且在整个大促（包括之前三年的“双11”大促）期间，数据库未发生过任何影响大促的重大故障。当然，这个结果，也得益于淘宝网整个应用架构的设计，以及这几年来革命性的闪存设备的迅猛发展。

2008年，淘宝DBA团队准备从Oracle转向MySQL的时候，团队中的大多数人对MySQL的了解都非常之少。当时国内技术圈对MySQL的讨论也不多见，网上能找到的大多数中文资料基本上关注的还是如何安装，如何配置主备复制等。而MySQL中文类的书籍，大部分还是和PHP放在一起，作为PHP开发中的一环来讲述的。所以当我们发现mysqperformanceblog.com这个相当专业的国外博客的时候，无不欣喜莫名。同时也知道了博客的作者们2008年出版的High Performance MySQL第二版（中文版于2010年1月出版），这本书被很多MySQL DBA们奉为圭臬，书的三位主要作者Baron Schwartz、Peter Zaitsev和Vadim Tkachenko也在MySQL DBA圈中耳熟能详，他们组建的Percona公司和Percona Server分支版本以及XtraDB存储引擎也逐渐为国内DBA所熟知。2011年12月，淘宝网和O'Reilly在北京联合举办的Velocity China 2011技术大会上，我们有幸邀请到Percona公司的华人专家季海东（目前已离职）来介绍MySQL 5.5 InnoDB/XtraDB的性能优化和诊断方法。在季海东先生的引导下，我们也和Peter通过Skype电话会议有过沟通，介绍了MySQL在淘宝的应用情况，我们对MySQL一些特性的需求，以及对MySQL做的一些patch，并随后保持了密切的邮件联系。有了这些铺垫，我们对于在生产系统中采用Percona Server 5.5也有了更大的信心，如今已有超过1000个Percona Server 5.5的实例在线上运行。所以今年上半年电子工业出版社的张春雨（侠少）编辑找到我来翻译本书的第三版的时候，很是激动，一口应承。

考虑到这么经典的书应该尽快地和读者见面，故此我邀请了团队中的MySQL专家周振兴（花名：苏普）、彭立勋、翟卫祥（花名：印风）、刘辉（花名：希羽）一起来翻译。其中，我负责前、推荐序和第1、2、3章，周振兴负责第5、6、7章，彭立勋负责第4、8、9、14章，翟卫祥负责第10、11、12、13章，刘辉负责第15、16章和附录部分，最后由我负责统稿。所以毫无疑问，这本书是团队合作的结晶。虽然我们满怀激情，但由于都是第一次参与翻译技术书籍，确实对困难有些预估不足，加上下半年为了准备“双11”等各种大促，需要在DBA团队满负荷的工作间隙挤出个人时间，初稿出来后，由于每个人翻译风格不太一致，几次审稿修订，也让本书的编辑李云静和白涛吃了不少苦头，在此对大家表示深深的感谢，是大家不懈的努力，才使得本书能够顺利地地和读者见面。但书中肯定还存在不少问题，恳请读者不吝指出，欢迎大家和我的新浪微博<http://weibo.com/NinGoo>进行互动。

同时还要感谢本书第二版的译者们，他们娴熟的语言技巧给了我们很多的参考。也要感谢帮助审稿的同事们，包括但并不仅限于张新铭（花名：俊达）、张瑞（花名：张瑞）、吴学章（花名：维西）等，彭立勋甚至还发动了他女朋友加入到审稿工作中，在此一并表示感谢。当然，最后还要感谢我的妻子Lala，在我占用了大量周末时间的时候能够给予支持，并承担了全部的家务，让我以译书为借口毫无心理负担地偷懒。

宁海元（花名：江枫）

2013年3月于余杭

目录

O'Reilly Media, Inc. 介绍
译者序
推荐序
前言
第1章 MySQL架构与历史
1.1 MySQL逻辑架构
1.1.1 连接管理与安全性
1.1.2 优化与执行
1.2 并发控制
1.2.1 读写锁
1.2.2 锁粒度
1.3 事务
1.3.1 隔离级别
1.3.2 死锁
1.3.3 事务日志
1.3.4 MySQL中的事务
1.4 多版本并发控制
1.5 MySQL的存储引擎
1.5.1 InnoDB存储引擎
1.5.2 MyISAM存储引擎
1.5.3 MySQL内建的其他存储引擎
1.5.4 第三方存储引擎
1.5.5 选择合适的引擎
1.5.6 转换表的引擎
1.6 MySQL时间线 (Timeline)
1.7 MySQL的开发模式
1.8 总结
第2章 MySQL基准测试
2.1 为什么需要基准测试
2.2 基准测试的策略
2.2.1 测试何种指标
2.3 基准测试方法
2.3.1 设计和规划基准测试
2.3.2 基准测试应该运行多长时间
2.3.3 获取系统性能和状态
2.3.4 获得准确的测试结果
2.3.5 运行基准测试并分析结果
2.3.6 绘图的重要性
2.4 基准测试工具
2.4.1 集成式测试工具
2.4.2 单组件式测试工具
2.5 基准测试案例
2.5.1 http_load
2.5.2 MySQL基准测试套件
2.5.3 sysbench
2.5.4 数据库测试套件中的dbt2/TPC-C测试
2.5.5 Percona的TPCC-MySQL测试工具
2.6 总结
第3章 服务器性能剖析
3.1 性能优化简介
3.1.1 通过性能剖析进行优化
3.1.2 理解性能剖析
3.2 对应用程序进行性能剖析
3.2.1 测量PHP应用程序
3.3 剖析MySQL查询
3.3.1 剖析服务器负载
3.3.2 剖析单条查询
3.3.3 使用性能剖析
3.4 诊断间歇性问题
3.4.1 单条查询问题还是服务器问题
3.4.2 捕获诊断数据
3.4.3 一个诊断案例
3.5 其他剖析工具
3.5.1 使用USER_STATISTICS表
3.5.2 使用strace
3.6 总结
第4章 Schema与数据类型优化
4.1 选择优化的数据类型
4.1.1 整数类型
4.1.2 实数类型
4.1.3 字符串类型
4.1.4 日期和时间类型
4.1.5 位数据类型
4.1.6 选择标识符 (identifier)
4.1.7 特殊类型数据
4.2 MySQL schema设计中的陷阱
4.3 范式和反范式
4.3.1 范式的优点和缺点
4.3.2 反范式的优点和缺点
4.3.3 混用范式和反范式化
4.4 缓存表和汇总表
4.4.1 物化视图
4.4.2 计数器表
4.5 加快ALTER TABLE操作的速度
4.5.1 只修改.frm文件
4.5.2 快速创建MyISAM索引
4.6 总结
第5章 创建高性能的索引
5.1 索引基础
5.1.1 索引的类型
5.2 索引的优点
5.3 高性能的索引策略
5.3.1 独立的列
5.3.2 前缀索引和索引选择性

- 5.3.3 多列索引
- 5.3.4 选择合适的索引列顺序
- 5.3.5 聚簇索引
- 5.3.6 覆盖索引
- 5.3.7 使用索引扫描来做排序
- 5.3.8 压缩（前缀压缩）索引
- 5.3.9 冗余和重复索引
- 5.3.10 未使用的索引
- 5.3.11 索引和锁
- 5.4 索引案例学习
- 5.4.1 支持多种过滤条件
- 5.4.2 避免多个范围条件
- 5.4.3 优化排序
- 5.5 维护索引和表
- 5.5.1 找到并修复损坏的表
- 5.5.2 更新索引统计信息
- 5.5.3 减少索引和数据的碎片
- 5.6 总结
- 第6章 查询性能优化
- 6.1 为什么查询速度会慢
- 6.2 慢查询基础：优化数据访问
- 6.2.1 是否向数据库请求了不需要的数据
- 6.2.2 MySQL是否在扫描额外的记录
- 6.3 重构查询的方式
- 6.3.1 一个复杂查询还是多个简单查询
- 6.3.2 切分查询
- 6.3.3 分解关联查询
- 6.4 查询执行的基础
- 6.4.1 MySQL客户端/服务器通信协议
- 6.4.2 查询缓存
- 6.4.3 查询优化处理
- 6.4.4 查询执行引擎
- 6.4.5 返回结果给客户端
- 6.5 MySQL查询优化器的局限性
- 6.5.1 关联子查询
- 6.5.2 UNION的限制
- 6.5.3 索引合并优化
- 6.5.4 等值传递
- 6.5.5 并行执行
- 6.5.6 哈希关联
- 6.5.7 松散索引扫描
- 6.5.8 最大值和最小值优化
- 6.5.9 在同一个表上查询和更新
- 6.6 查询优化器的提示（hint）
- 6.7 优化特定类型的查询
- 6.7.1 优化COUNT()查询
- 6.7.2 优化关联查询
- 6.7.3 优化子查询
- 6.7.4 优化GROUP BY和DISTINCT
- 6.7.5 优化LIMIT分页
- 6.7.6 优化SQL_CALC_FOUND_ROWS
- 6.7.7 优化UNION查询
- 6.7.8 静态查询分析
- 6.7.9 使用用户自定义变量
- 6.8 案例学习
- 6.8.1 使用MySQL构建一个队列表
- 6.8.2 计算两点之间的距离
- 6.8.3 使用用户自定义函数
- 6.9 总结
- 第7章 MySQL高级特性
- 7.1 分区表
- 7.1.1 分区表的原理
- 7.1.2 分区表的类型
- 7.1.3 如何使用分区表
- 7.1.4 什么情况下会出问题
- 7.1.5 查询优化
- 7.1.6 合并表
- 7.2 视图
- 7.2.1 可更新视图
- 7.2.2 视图对性能的影响
- 7.2.3 视图的限制
- 7.3 外键约束
- 7.4 在MySQL内部存储代码
- 7.4.1 存储过程和函数
- 7.4.2 触发器
- 7.4.3 事件
- 7.4.4 在存储程序中保留注释
- 7.5 游标
- 7.6 绑定变量
- 7.6.1 绑定变量的优化
- 7.6.2 SQL接口的绑定变量
- 7.6.3 绑定变量的限制
- 7.7 用户自定义函数
- 7.8 插件
- 7.9 字符集和校对
- 7.9.1 MySQL如何使用字符集
- 7.9.2 选择字符集和校对规则
- 7.9.3 字符集和校对规则如何影响查询
- 7.10 全文索引
- 7.10.1 自然语言的全文索引
- 7.10.2 布尔全文索引
- 7.10.3 MySQL 5.1中全文索引的变化
- 7.10.4 全文索引的限制和替代方案
- 7.10.5 全文索引的配置和优化
- 7.11 分布式（XA）事务
- 7.11.1 内部XA事务
- 7.11.2 外部XA事务

- 7.12 查询缓存
 - 7.12.1 MySQL如何判断缓存命中
 - 7.12.2 查询缓存如何使用内存
 - 7.12.3 什么情况下查询缓存能发挥作用
 - 7.12.4 如何配置和维护查询缓存
 - 7.12.5 InnoDB和查询缓存
 - 7.12.6 通用查询缓存优化
 - 7.12.7 查询缓存的替代方案
- 7.13 总结
- 第8章 优化服务器设置
 - 8.1 MySQL配置的工作原理
 - 8.1.1 语法、作用域和动态性
 - 8.1.2 设置变量的副作用
 - 8.1.3 入门
 - 8.1.4 通过基准测试迭代优化
 - 8.2 什么不该做
 - 8.3 创建MySQL配置文件
 - 8.3.1 检查MySQL服务器状态变量
 - 8.4 配置内存使用
 - 8.4.1 MySQL可以使用多少内存
 - 8.4.2 每个连接需要的内存
 - 8.4.3 为操作系统保留内存
 - 8.4.4 为缓存分配内存
 - 8.4.5 InnoDB缓冲池 (Buffer Pool)
 - 8.4.6 MyISAM键缓存 (Key Caches)
 - 8.4.7 线程缓存
 - 8.4.8 表缓存 (Table Cache)
 - 8.4.9 InnoDB数据字典 (Data Dictionary)
 - 8.5 配置MySQL的I/O行为
 - 8.5.1 InnoDB I/O配置
 - 8.5.2 MyISAM的I/O配置
 - 8.6 配置MySQL并发
 - 8.6.1 InnoDB并发配置
 - 8.6.2 MyISAM并发配置
 - 8.7 基于工作负载的配置
 - 8.7.1 优化BLOB和TEXT的场景
 - 8.7.2 优化排序 (Filesorts)
 - 8.8 完成基本配置
 - 8.9 安全和稳定的设置
 - 8.10 高级InnoDB设置
 - 8.11 总结
- 第9章 操作系统和硬件优化
 - 9.1 什么限制了MySQL的性能
 - 9.2 如何为MySQL选择CPU
 - 9.2.1 哪个更好：更快的CPU还是更多的CPU
 - 9.2.2 CPU架构
 - 9.2.3 扩展到多个CPU和核心
 - 9.3 平衡内存和磁盘资源
 - 9.3.1 随机I/O和顺序I/O
 - 9.3.2 缓存，读和写
 - 9.3.3 工作集是什么
 - 9.3.4 找到有效的内存/磁盘比例
 - 9.3.5 选择硬盘
 - 9.4 固态存储
 - 9.4.1 闪存概述
 - 9.4.2 闪存技术
 - 9.4.3 闪存的基准测试
 - 9.4.4 固态硬盘驱动器 (SSD)
 - 9.4.5 PCIe存储设备
 - 9.4.6 其他类型的固态存储
 - 9.4.7 什么时候应该使用闪存
 - 9.4.8 使用Flashcache
 - 9.4.9 优化固态存储上的MySQL
 - 9.5 为备库选择硬件
 - 9.6 RAID性能优化
 - 9.6.1 RAID的故障转移、恢复和镜像
 - 9.6.2 平衡硬件RAID和软件RAID
 - 9.6.3 RAID配置和缓存
 - 9.7 SAN和NAS
 - 9.7.1 SAN基准测试
 - 9.7.2 使用基于NFS或SMB的SAN
 - 9.7.3 MySQL在SAN上的性能
 - 9.7.4 应该用SAN吗
 - 9.8 使用多磁盘卷
 - 9.9 网络配置
 - 9.10 选择操作系统
 - 9.11 选择文件系统
 - 9.12 选择磁盘队列调度策略
 - 9.13 线程
 - 9.14 内存交换区
 - 9.15 操作系统状态
 - 9.15.1 如何阅读vmstat的输出
 - 9.15.2 如何阅读iostat的输出
 - 9.15.3 其他有用的工具
 - 9.15.4 CPU密集型的机器
 - 9.15.5 I/O密集型的机器
 - 9.15.6 发生内存交换的机器
 - 9.15.7 空闲的机器
 - 9.16 总结
- 第10章 复制
 - 10.1 复制概述
 - 10.1.1 复制解决的问题
 - 10.1.2 复制如何工作
 - 10.2 配置复制
 - 10.2.1 创建复制账号
 - 10.2.2 配置主库和备库
 - 10.2.3 启动复制

- [10.2.4 从另一个服务器开始复制](#)
- [10.2.5 推荐的复制配置](#)
- [10.3 复制的原理](#)
 - [10.3.1 基于语句的复制](#)
 - [10.3.2 基于行的复制](#)
 - [10.3.3 基于行或基于语句：哪种更优](#)
 - [10.3.4 复制文件](#)
 - [10.3.5 发送复制事件到其他备库](#)
 - [10.3.6 复制过滤器](#)
- [10.4 复制拓扑](#)
 - [10.4.1 一主库多备库](#)
 - [10.4.2 主动-主动模式下的主-主复制](#)
 - [10.4.3 主动-被动模式下的主-主复制](#)
 - [10.4.4 拥有备库的主-主结构](#)
 - [10.4.5 环形复制](#)
 - [10.4.6 主库、分发主库以及备库](#)
 - [10.4.7 树或金字塔形](#)
 - [10.4.8 定制的复制方案](#)
- [10.5 复制和容量规划](#)
 - [10.5.1 为什么复制无法扩展写操作](#)
 - [10.5.2 备库什么时候开始延迟](#)
 - [10.5.3 规划冗余容量](#)
- [10.6 复制管理和维护](#)
 - [10.6.1 监控复制](#)
 - [10.6.2 测量备库延迟](#)
 - [10.6.3 确定主备是否一致](#)
 - [10.6.4 从主库重新同步备库](#)
 - [10.6.5 改变主库](#)
 - [10.6.6 在一个主-主配置中交换角色](#)
- [10.7 复制的问题和解决方案](#)
 - [10.7.1 数据损坏或丢失的错误](#)
 - [10.7.2 使用非事务型表](#)
 - [10.7.3 混合事务型和非事务型表](#)
 - [10.7.4 不确定语句](#)
 - [10.7.5 主库和备库使用不同的存储引擎](#)
 - [10.7.6 备库发生数据改变](#)
 - [10.7.7 不唯一的服务器ID](#)
 - [10.7.8 未定义的服务器ID](#)
 - [10.7.9 对未复制数据的依赖性](#)
 - [10.7.10 丢失的临时表](#)
 - [10.7.11 不复制所有的更新](#)
 - [10.7.12 InnoDB加锁读引起的锁争用](#)
 - [10.7.13 在主-主复制结构中写入两台主库](#)
 - [10.7.14 过大的复制延迟](#)
 - [10.7.15 来自主库的过大的包](#)
 - [10.7.16 受限制的复制带宽](#)
 - [10.7.17 磁盘空间不足](#)
 - [10.7.18 复制的局限性](#)
- [10.8 复制有多快](#)
- [10.9 MySQL复制的高级特性](#)
- [10.10 其他复制技术](#)
- [10.11 总结](#)
- [第11章 可扩展的MySQL](#)
 - [11.1 什么是可扩展性](#)
 - [11.1.1 正式的可扩展性定义](#)
 - [11.2 扩展MySQL](#)
 - [11.2.1 规划可扩展性](#)
 - [11.2.2 为扩展赢得时间](#)
 - [11.2.3 向上扩展](#)
 - [11.2.4 向外扩展](#)
 - [11.2.5 通过多实例扩展](#)
 - [11.2.6 通过集群扩展](#)
 - [11.2.7 向内扩展](#)
 - [11.3 负载均衡](#)
 - [11.3.1 直接连接](#)
 - [11.3.2 引入中间件](#)
 - [11.3.3 一主多备间的负载均衡](#)
 - [11.4 总结](#)
- [第12章 高可用性](#)
 - [12.1 什么是高可用性](#)
 - [12.2 导致宕机的原因](#)
 - [12.3 如何实现高可用性](#)
 - [12.3.1 提升平均失效时间 \(MTBF\)](#)
 - [12.3.2 降低平均恢复时间 \(MTTR\)](#)
 - [12.4 避免单点失效](#)
 - [12.4.1 共享存储或磁盘复制](#)
 - [12.4.2 MySQL同步复制](#)
 - [12.4.3 基于复制的冗余](#)
 - [12.5 故障转移和故障恢复](#)
 - [12.5.1 提升备库或切换角色](#)
 - [12.5.2 虚拟IP地址或IP接管](#)
 - [12.5.3 中间件解决方案](#)
 - [12.5.4 在应用中处理故障转移](#)
 - [12.6 总结](#)
- [第13章 云端的MySQL](#)
 - [13.1 云的优点、缺点和相关误解](#)
 - [13.2 MySQL在云端的经济价值](#)
 - [13.3 云中的MySQL的可扩展性和高可用性](#)
 - [13.4 四种基础资源](#)
 - [13.5 MySQL在云主机上的性能](#)
 - [13.5.1 在云端的MySQL基准测试](#)
 - [13.6 MySQL数据库即服务 \(DBaaS\)](#)
 - [13.6.1 Amazon RDS](#)
 - [13.6.2 其他DBaaS解决方案](#)
 - [13.7 总结](#)
- [第14章 应用层优化](#)
 - [14.1 常见问题](#)

- 14.2 [Web服务器问题](#)
- 14.2.1 [寻找最优并发度](#)
- 14.3 [缓存](#)
- 14.3.1 [应用层以下的缓存](#)
- 14.3.2 [应用层缓存](#)
- 14.3.3 [缓存控制策略](#)
- 14.3.4 [缓存对象分层](#)
- 14.3.5 [预生成内容](#)
- 14.3.6 [作为基础组件的缓存](#)
- 14.3.7 [使用HandlerSocket和memcached](#)
- 14.4 [拓展MySQL](#)
- 14.5 [MySQL的替代品](#)
- 14.6 [总结](#)
- 第15章 [备份与恢复](#)
- 15.1 [为什么要备份](#)
- 15.2 [定义恢复需求](#)
- 15.3 [设计MySQL备份方案](#)
- 15.3.1 [在线备份还是离线备份](#)
- 15.3.2 [逻辑备份还是物理备份](#)
- 15.3.3 [备份什么](#)
- 15.3.4 [存储引擎和一致性](#)
- 15.4 [管理和备份二进制日志](#)
- 15.4.1 [二进制日志格式](#)
- 15.4.2 [安全地清除老的二进制日志](#)
- 15.5 [备份数据](#)
- 15.5.1 [生成逻辑备份](#)
- 15.5.2 [文件系统快照](#)
- 15.6 [从备份中恢复](#)
- 15.6.1 [恢复物理备份](#)
- 15.6.2 [还原逻辑备份](#)
- 15.6.3 [基于时间点的恢复](#)
- 15.6.4 [更高级的恢复技术](#)
- 15.6.5 [InnoDB崩溃恢复](#)
- 15.7 [备份和恢复工具](#)
- 15.7.1 [MySQL Enterprise Backup](#)
- 15.7.2 [Percona XtraBackup](#)
- 15.7.3 [mylmbbackup](#)
- 15.7.4 [Zmanda Recovery Manager](#)
- 15.7.5 [mysdumper](#)
- 15.7.6 [mysqldump](#)
- 15.8 [备份脚本化](#)
- 15.9 [总结](#)
- 第16章 [MySQL用户工具](#)
- 16.1 [接口工具](#)
- 16.2 [命令行工具集](#)
- 16.3 [SQL实用集](#)
- 16.4 [监测工具](#)
- 16.4.1 [开源的监控工具](#)
- 16.4.2 [商业监控系统](#)
- 16.4.3 [Innotop的命令行监控](#)
- 16.5 [总结](#)
- 附录A [MySQL分支与变种](#)
- 附录B [MySQL服务器状态](#)
- 附录C [大文件传输](#)
- 附录D [EXPLAIN](#)
- 附录E [锁的调试](#)
- 附录F [在MySQL上使用Sphinx](#)
- [索引](#)

推荐序

很多年前我就是这本书的“粉丝”了，这是一本伟大的书，第三版尤其如此。这些世界级的专家不仅仅分享他们的专业知识，也花了很多时间来更新和添加新的章节，且都是高品质的内容。本书有大量关于如何获得MySQL高性能的细节信息，并且关注的是提升性能的过程，而不仅仅是描述事实结果和琐碎的细枝末节。这本书将告诉读者如何将事情做得更好，不管MySQL在不同版本中的行为有多么大的改变。

毫无疑问，本书的作者是唯一有资格来写这么一本书的人，他们经验丰富，有合理的方法，关注效率，并且精益求精。说到经验丰富，本书的作者已经在MySQL性能领域工作多年，从MySQL还没有什么可扩展性和可测量性的时代，直到现在这些方面已经有了长足的进步。而说到合理的方法，他们简直把这件事情当成了科学，首先定义需要解决的问题，然后通过合理的猜测和精确的测量来解决问题。

我对作者在效率方面的关注尤其印象深刻。作为顾问，他们时间宝贵。客户是按照他们的时间付费的，所以都希望能更快地解决问题。所以本书作者定义了一整套的流程，开发了很多的工具，让事情变得正确和高效。在本书中，作者详细描述了这些流程，并且发布了工具的源代码。

最后，本书作者在工作上一一直精益求精。比如从吞吐量到响应时间的关注，致力于了解MySQL在新硬件上的性能表现，追求新的技能如排队理论对性能的影响，等等。

我相信本书预示了MySQL的光明前景。MySQL已经支持高要求的工作负载，本书作者也在努力提升MySQL社区内对性能的认识。同时，他们还直接为性能提升做出了贡献，包括XtraDB和XtraBackup。一直以来我从他们身上学到了不少东西，也希望读者多花点时间读读本书，一定会同样有所收益。

——Mark Callaghan, Facebook软件工程师

欢迎访问：电子书学习和下载网站 (<https://www.shgis.cn>)

文档名称：《高性能MySQL：第3版》施瓦茨 (Baron Schwartz) 著.epub

请登录 <https://shgis.cn/post/332.html> 下载完整文档。

手机端请扫码查看：

