<<高性能MySQL>>

图书基本信息

书名：<<高性能MySQL>>

13位ISBN编号：9787564134457

10位ISBN编号：7564134453

出版时间：2012-6

出版时间：东南大学出版社

作者：（美）施瓦兹，（美）扎伊采夫，（美）特卡申科　著

页数：793

字数：1013000

前言

We wrote this book to serve the needs of not just the MySQL application developer but also the MySQL database administrator. We assume that you are already relatively experienced with MySQL. We also assume some experience with general system ad ministration, networking, and Unix-like operating systems. The second edition of this book presented a lot of information to readers, but no book can provide complete coverage of a topic. Between the second and third editions, we took notes, on literally thousands of interesting problems we'd solved or seen others solve. When we started to outline the third edition, it became clear that not only would full coverage of these topics require three to five thousand pages, but the book still wouldn't be complete. After reflecting on this problem, we realized that the second edition's emphasis on deep coverage was actually self-limiting, in the sense that it often didn't teach readers how to think about MySQL. As a result, this third edition has a different focus from the second edition. We still convey a lot of information, and we still emphasize the same goals, such as reliability and correctness. But we've also tried to imbue the book with a deeper purpose: we want fo teach the principles of why MySQL works as it does, not just the facts about how it works. We've included more illustrative stories and case studies, which demonstrate the principles in action. We build on these to try to answer questions such as "Given MySQL's internal architecture and operation, what practical effects arise in real usage?Why do those effects matter? How do they make MySQL well suited （or not well suited） for particular needs?" Ultimately, we hope that your knowledge of MySQL's internals will help you in situa tions beyond the scope of this book. And we hope that your newfound insight will help you to learn and practice a methodical approach to designing, maintaining, and trou bleshooting systems that are built on MySQL. How This Book Is Organized We fit a lot of complicated topics into this book. Here, we explain how we put them together in an order that makes them easier to learn. A Broad Overview Chapter 1, MySQL Architecture and History is dedicated to the basics--things you'll need to be familiar with before you dig in deeply. You need to understand how MySQL is organized before you'll be able to use it effectively. This chapter explains MySQL's architecture and key facts about its storage engines. It helps you get up to speed if you aren't familiar with some of the fundamentals of a relational database, including transactions. This chapter will also be useful if this book is your introduction to MySQL but you're already familiar with another database, such as Oracle. We also include a bit of historical context: the changes to'MySQL over time, recent ownership changes, and where we think it's headed. Building a Solid Foundation The early chapters cover material we hope you'll reference over and over as you use MySQL. Chapter 2, Benchmarking MySQL discusses the basics of benchmarking--that is, determining what sort of workload your server can handle, how fast it can perform certain tasks, and so on. Benchmarking is an essential skill for evaluating how the server behaves under load, but it's also important to know when it's not useful.

Chapter 3, Profiling Server Performance introduces you to the response time-oriented approach we take to troubleshooting and diagnosing server performance problems.This framework has proven essential to solving some of the most puzzling cases we've seen. Although you might choose to modify our approach （we developed it by modilying Cary Millsap's approach, after all）, we hope you'll avoid the pitfalls of not having any method at all.

In Chapters 4 through 6, we introduce three topics that together form the foundation for a good logical and physical database design. In Chapter 4, Optimizing Schema and Data Types, we cover the various nuances of data types and table design. Chapter 5,Indexing for High Performance extends the discussion to indexes--that is, physical database design. A firm understanding of indexes and how to use them well is essential for using MySQL effective'y, so you'll probably find yourself returning to this chapter repeatedly. And Chapter 6, Query Performance Optimization wraps the topics together by explaining how MySQL executes queries and how you can take advantage of its query optimizer's strengths. This chapter also presents specific examples of many Common classes of queries, illustrating where MySQL does a good job and how to transform queries into forms that use its strengths. Up to this point, we've covered the basic topics that apply to any database: tables,indexes, data, and queries. Chapter 7, Advanced MySQL Features goes beyond the basics and shows you how MySQL's advanced features work. We examine topics such as partkioning, storedprocedures, triggers, and character sets. MySQL's

implementation of these features is different from other databases, and a good-understanding of them can open up new opportunities for performance gains that you might not have thought about otherwise. Configuring Your Application The next two chapters discuss how to make MySQL, your application, and your hard ware workwell together. In Chapter 8, Optimizing Server Settings, we discuss how you can configure MySQL to make the most of your hardware and to be reliable and robust. Chapter 9, Operating System and Hardware Optimization explains how to get the most out of your operating system and hardware. We discuss solid-state storage in depth, and we suggest hardware configurations that might provide better performance for larger-scale applications.

Both chapters explore MySQL internals to some degree. This is a recurring theme that continues all the way through the appendixes: learn how it works internally, and you'll be empowered to understand and reason about the consequences: MySQL as an Infrastructure Component MySQL doesn't exist in a vacuum. It's part of an overall application stack, and you'll need to build a robust overall architecture for your application. The next set of chapters is about how to do that. In Chapter 10, Replication, we discuss MySQL's killer feature: the ability to set up multiple servers that all stay in sync with a master server's changes. Unfortunately,replication is perhaps MySQL's most troublesome feature for some people. This doesn't have to be the case, and we show you how to ensure that it keeps runni.ng well.Chapter 11, Scaling MySQL discusses what scalability is （it's not the same thing as performance）, why applications and systems don't scale, and what to do about it. If you do it right, you can scale MySQL to suit nearly any purpose. Chapter 12, High Availability delves into a related-but-distinct topic: how to ensure that MySQL stays hp and functions smoothly. In Chapter 13, MySQL in the Cloud, you'll learn about what's different when you run MySQL in cloud computing environments.

<<高性能MySQL>>

内容概要

本书中，你将学到与MySQL数据库模式、索引和查询设计相关的所有高级技巧。
通过使用这些技巧，你将能够对。
MySQL数据库服务器、操作系统和硬件进行调优以发挥它们的最大潜力。
这本指南同时也提供了通过复制、负载均衡、高可用性及故障转移等技术对应用进行扩展的安全实用方法。

《高性能MySQL(影印版第3版)由Baron Schwartz、Peter Zaitsev、Vadim Tkachenko著。
在第三版中更新了MysQL数据库和InnoDB存储引擎在性能、特性和工具等相关领域取得的最新进展，不仅通过大量的特定示例讲解了MySQL数据库如何工作，同时也围绕MysQL数据库的设计原则，采用生动的故事和案例研究的形式解释了其对应的工作原理。
本书教给读者如何采用MySQL的方式进行思考。

<<高性能MySQL>>

作者简介

作者：（美国）施瓦兹（Baron Scbwartz）（美国）扎伊采夫（Peter Zaitsev）（美国）特卡申科（Vadim Tkacbenko） 施瓦兹（Schwartz B.），是Percona公司的首席性能架构师。
他的主要工作是通过创建一系列工具和技术来提高MySQL数据库的易用性和可靠性。
扎伊采夫（zaitsev P.），是Percona公司的首席执行官和联合创始人之一。
作为一名数据库内核、计算机硬件和应用扩展技术领域的专家，他管理着MysQL团队内部的高性能技术组直到2006年。
特卡申科（Tkachenko V.）是Percona公司的首席技术官和联合创始人之一。
他领导着公司的开发团队先后开发了Percona服务器、Percona XtraDB集群和Percona XtraBackup工具等产品。

<<高性能MySQL>>

书籍目录

章节摘录

版权页： 插图： Too many columns MySQL's storage engine API works by copying rows between the server and the storage engine in a row buffer format;the server then decodes the buffer into col-umns.But it can be costly to turn the row buffer into the row data structure with the decoded columns.MylSAM's fixed row format actually matches the server's row format exactly,so no conversion is needed.However,MylSAM's variable row format and InnoDB's row format always require conversion.The cost of this con-version depends on the number of columns.We discovered that this can become expensive when we investigated an issue with high CPU consumption for a cus-tomer with extremely wide tables (hundreds of columns),even though only a few columns were actually used.If you're planning for hundreds of columns,be aware that the server's performance characteristics will be a bit different. Too many joins The so-called entity-attribute-value (EAV) design pattern is a classic case of a uni-versally bad design pattern that especially doesn't work well in MySQL.MySQL has a limitation of 61 tables per join,and EAV databases require many self-joins. We've seen more than a few EAV databases eventually exceed this limit.Even at many fewer joins than 61,however,the cost of planning and optimizing the query can become problematic for MySQL.As a rough rule of thumb,it's better to have a dozen or fewer tables per query if you need queries to execute very fast with high concurrency. The all-powerful ENUM Beware of overusing ENUM.Here's an example we saw： CREATE TABLE ...(country enum('','0','1','2',...,'31') The schema was sprinkled liberally with this pattern.This would probably be a questionable design decision in any database with an enumerated value type,be-cause it really should be an integer that is foreign-keyed to a "dictionary" or "lookup" table anyway.But in MySQL,you can't add a new country to the list without an ALTER TABLE,which is a blocking operation in MySQL 5.0 and earlier,and even in 5.1 and newer if you add the value anywhere but at the end of the list.

媒体关注与评论

“同本书的前几版相比，第三版是一本更加棒的书。
这几位作者是唯一具有资格写这本书的人。
我将继续向他们学习，并希望你也能抽出时间这样做。
”　　——Mark Callaghan Facebook软件工程师

版权说明