HZ BOOKS
华章科技

从源代码角度深度解析InnoDB的体系结构、实现原理和工作机制

资深MySQL专家亲自执笔，全球知名MySQL数据库服务提供商Percona公司CTO作序推荐

数据库技术丛书

Inside MySQL: InnoDB Storage Engine

MySQL技术内幕

InnoDB存储引擎

姜承尧◎著

机械工业出版社
China Machine Press

数据库 技术丛书

MySQL技术内幕
InnoDB存储引擎

姜承尧◎著

机械工业出版社
China Machine Press

本书是国内目前唯一的一本关于InnoDB的著作，由资深MySQL专家亲自执笔，中外数据库专家联袂推荐，权威性毋庸置疑。

内容深入，从源代码的角度深度解析了InnoDB的体系结构、实现原理、工作机制，并给出了大量最佳实践，能帮助你系统而深入地掌握InnoDB，更重要的是，它能为你设计和管理高性能、高可用的数据库系统提供绝佳的指导。注重实战，全书辅有大量的案例，可操作性极强。

全书首先全景式地介绍了MySQL独有的插件式存储引擎，分析了MySQL的各种存储引擎的优势和应用环境；接着以InnoDB的内部实现为切入点，逐一详细讲解了InnoDB存储引擎内部的各个功能模块，包括InnoDB存储引擎的体系结构、内存中的数据结构、基于InnoDB存储引擎的表和页的物理存储、索引与算法、文件、锁、事务、备份，以及InnoDB的性能调优等重要的知识；最后深入解析了InnoDB存储引擎的源代码结构，对大家阅读和理解InnoDB的源代码有重要的指导意义。

本书适合所有希望构建和管理高性能、高可用性的MySQL数据库系统的开发者和DBA阅读。

图书在版编目（CIP）数据

MySQL技术内幕：InnoDB存储引擎 / 姜承尧著. —北京：机械工业出版社，2010.11

ISBN 978-7-111-32188-0

Ⅰ. M…　Ⅱ. 姜…　Ⅲ. 关系数据库-数据库管理系统，MySQL　Ⅳ. TP311.138

中国版本图书馆CIP数据核字（2010）第198251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：陈佳媛
北京市荣盛彩色印刷有限公司印刷.
2011年1月第1版第1次印刷
186mm × 240 mm · 25.25印张
标准书号：ISBN 978-7-111-32188-0
定价：69.00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

推 荐 序

It's fair to say that MySQL is the most popular open source database. It has a very large installed base and number of users. Let's see what are the reasons MySQL is so popular, where it stands currently, and maybe touch on some of its future (although predicting the future is rarely successful).

Looking at the customer area of MySQL, which includes Facebook, Flickr, Adobe (in Creative Suite 3), Drupal, Digg, LinkedIn, Wikipedia, eBay, YouTube, Google AdSense (source http://mysql.com/customers/ and public resources), it's obvious that MySQL is everywhere. When you log in to your popular forum (powered by Bulleting) or blog (powered by WordPress), most likely it has MySQL as its backend database. Traditionally, two MySQL's characteristics, simplicity of use and performance, were what allowed it to gain such popularity. In addition to that, availability on a very wide range of platforms (including Windows) and built-in replication, which provides an easy scale-out solution for read-only clients, gave more user attractions and production deployments. There is simple evidence of MySQL's simplicity: In 15 minutes or less, you really can get installed, have a working database, and start running queries and store data. From its early stages MySQL had a good interface to most popular languages for Web development - PHP and Perl, and also Java and ODBC connectors.

There are two best known storage engines in MySQL: MyISAM and InnoDB (I don't cover NDB cluster here; it's a totally different story). MyISAM comes as the default storage engine and historically it is the oldest, but InnoDB is ACID compliant and provides transactions, row-level locking, MVCC, automatic recovery and data corruption detection. This makes it the storage engine you want to choose for your application. Also, there is the third-party transaction storage engine PBXT, with characteristics similar to InnoDB, which is included in the MariaDB distribution.

MySQL's simplicity has its own drawback. Just as it is very easy to start working with it, it is very easy to start getting into trouble with it. As soon as your website or forum gets popular, you

may figure out that the database is a bottleneck, and that you need special skills and tools to fix it.

The author of this book is a MySQL expert, especially in InnoDB storage engineB. Hence, I highly recommend this book to new users of InnoDB as well as uers who already have well-tuned InnoDB-based applications but need to get internal out of them.

Vadim Tkachenko

全球知名MySQL数据库服务提供商Percona公司CTO

知名MySQL数据库博客MySQLPerformanceBlog.com作者

《高性能MySQL（第2版）》作者之一

前　言

过去这些年，我一直在和各种不同的数据库打交道，见证了MySQL从一个小型的关系型数据库发展成为各大企业的核心数据库系统的过程，并且参与了一些大大小小的项目的开发工作，成功地帮助开发人员构建了一些可靠、健壮的应用程序。在这个过程中积累了一些经验，正是这些不断累积的经验赋予了我灵感，于是有了本书。这本书实际上反映了这些年来我做了哪些事情，汇集了很多同行每天可能都会遇到的一些问题，并给出了解决方案。

MySQL数据库独有的插件式存储引擎架构使得它与其他任何数据库都不同，不同的存储引擎有着完全不同的功能，而InnoDB存储引擎的存在使得MySQL跃入了企业级数据库领域。**本书完整地讲解了InnoDB存储引擎中最重要的一些内容，即InnoDB的体系结构和工作原理，并结合InnoDB的源代码讲解了它的内部实现机制。**

本书不仅介绍了InnoDB存储引擎的诸多功能和特性，而且还阐述了如何正确地使用这些功能和特性。更重要的是，它还尝试教大家如何Think Different。Think Different是20世纪90年代苹果公司在其旷日持久的宣传活动中提出的一个口号，借此来重振公司的品牌，更重要的是改变人们对技术在日常生活中的作用的看法。需要注意的是，苹果的口号不是Think Differently，而是Think Different。这里的Different是名词，意味该思考些什么。

很多DBA和开发人员都相信某些“神话”，然而这些“神话”往往都是错误的。无论计算机技术发展的速度变得多快、数据库的使用变得多么简单，任何时候WHY都比WHAT重要。只有真正地理解了内部实现原理、体系结构，才能更好地去使用。这正是人类正确思考问题的原则。因此，对于当前出现的技术，尽管学习应用层面的技术很重要，但更重要的是，应当正确地理解和使用这些技术。

关于这本书，我想实现好几个目标，但最重要的是想告诉大家如下几个简单的观点：

- ❑ 不要相信任何“神话”，学会自己思考。
- ❑ 不要墨守成规，大部分人都知道的事情可能是错误的。
- ❑ 不要相信网上的传言，去测试，根据自己的实践做出决定。
- ❑ 花时间充分地思考，敢于提出质疑。

为什么写本书

当前有关MySQL的书籍大部分都集中在教读者如何使用MySQL，例如SQL语句的使用、复制的搭建、数据的切分等。没错，这对快速掌握和使用MySQL数据库非常有好处，但是真正的数据库工作者需要了解的不仅仅是应用，更多的是内部的具体实现。

MySQL数据库独有的插件式存储引擎结构使得想要在一本书内完整地讲解各个存储引擎变得十分困难。有的书可能偏重于对MyISAM的介绍，有的书则可能偏重于对InnoDB存储引擎的介绍。对于初级的DBA来说，这可能会使他们的理解变得更困难。对于大多数MySQL DBA和开发人员来说，他们往往更希望了解作为MySQL企业级数据库应用的第一存储引擎——InnoDB。我想在本书中，他们可以找到他们想要的内容。

再强调一遍，任何时候WHY都比WHAT重要。本书从源代码的角度对InnoDB的存储引擎的整个体系架构的各个组成部分进行了系统的分析和讲解，剖析了InnoDB存储引擎的核心实现和工作机制，相信这在其他书中是很难找到的。

本书面向的读者

本书不是一本面向应用的数据库类书籍，也不是一本参考手册，更不会教你如何在MySQL中使用SQL语句。本书面向那些使用MySQL InnoDB存储引擎作为数据库后端开发应用程序的开发者和有一定经验的MySQL DBA。书中的大部分例子都是用SQL语句来展示关键特性的，如果想通过本书来了解如何启动MySQL，如何配置Replication环境，可能并不能如愿。不过，通过本书，你将理解InnoDB存储引擎是如何工作的，它的关键特性的功能和作用是什么，以及如何正确地配置和使用这些特性。

如果想更好地使用InnoDB存储引擎，如果想让你的数据库应用获得更好的性能，就请阅读本书。从某种程度上讲，技术经理或总监也要非常了解数据库，要知道数据库对于企业的重要性。如果技术经理或总监想安排员工参加MySQL数据库技术方面的培训，完全可以利用本书来“充电”，相信你一定不会失望的。

要想更好地学习本书，要求具备以下条件：

- 掌握SQL。
- 掌握基本的MySQL操作。
- 接触过一些高级语言，如C、C++、Python或Java。
- 对一些基本算法有所了解，因为本书会分析InnoDB存储引擎的部分源代码，如果你能看懂这些代码，这会对你的理解非常有帮助。

如何阅读本书

本书一共有10章，每一章都像一本“迷你书”，可以单独成册，也就说，你完全可以从书中任何一章开始阅读。例如，要了解第10章中的InnoDB源代码编译和调试的知识，就不必先去阅读第3章有关文件的知识。当然，如果你不太确定自己是否已经对本书所涉及的内容已经完全掌握，建议你系统地阅读本书。

本书不是一本入门书，不会一步步引导你去如何操作。倘若你尚不了解InnoDB存储引擎，本书对你来说可能就显得沉重了些，建议你先查阅官方的API文档，大致掌握InnoDB的基础知识，然后再来阅读本书，相信你会领略到不同的风景。

需要特别说明的是，附录B～D详细给出了Master Thread、Doublewrite、哈希算法和哈希表的源代码，前面学习了这些理论知识，再适当地阅读这些源代码，相信有经验的DBA可以更好地掌握和理解InnoDB存储引擎的本质。为了便于大家阅读，本书在提供源代码下载的同时也将源代码附在了书中，因此占去了一些篇幅，还请大家理解。

致　谢

在编写本书的过程中，我得到了很多朋友的热心帮助。首先要感谢Pecona公司的CEO Peter Zaitsev和CTO Vadim Tkachenko，通过与他们的不断交流，使得我对InnoDB存储引擎有了更进一步的了解，同时知道了怎样才能正确地将InnoDB存储引擎的补丁应用到生产环境。

其次，我要感谢久游网公司的各位同事们。能在才华横溢、充满创意的团队中工作，我感到非常荣幸和兴奋。也因为在这个开放的工作环境中，我才得以不断地进行研究和创新。

此外，我还要感谢我的母亲，写书不是一件容易的事，特别是本书还想传达一些思想，在这个过程中我遇到了很多的困难，感谢她在这个过程中给予我的支持和鼓励。

最后，一份特别的感谢要送给本书的策划编辑杨福川先生和曾珊女士，他们使得本书变得生动和更具有灵魂。

目　录

第1章 MySQL体系结构和存储引擎

MySQL被设计为一个可移植的数据库，几乎能在当前所有的操作系统上运行，如Linux、Solaris、FreeBSD、Mac和Windows。尽管各种系统在底层（如线程）实现方面各有不同，但是MySQL几乎能保证在各平台上的物理体系结构的一致性。所以，你应该能很好地理解MySQL在所有这些平台上是如何工作的。

1.1 定义数据库和实例

在数据库领域中有两个词很容易混淆，它们就是“实例”（instance）和“数据库”（database）。作为常见的数据库术语，这两个词的定义如下。

- 数据库：物理操作系统文件或其他形式文件类型的集合。在MySQL中，数据库文件可以是frm、myd、myi、ibd结尾的文件。当使用NDB引擎时，数据库的文件可能不是操作系统上的文件，而是存放于内存之中的文件，但是定义仍然不变。
- 数据库实例：由数据库后台进程/线程以及一个共享内存区组成。共享内存可以被运行的后台进程/线程所共享。需要牢记的是，数据库实例才是真正用来操作数据库文件的。

这两个词有时可以互换使用，但两者的概念完全不同。在MySQL中，实例和数据库的通常关系是一一对应，即一个实例对应一个数据库，一个数据库对应一个实例。但是，在集群情况下可能存在一个数据库可被多个实例使用的情况。

MySQL被设计为一个单进程多线程架构的数据库，这点与SQL Server比较类似，但与Oracle多进程的架构有所不同（Oracle的Windows版本也是单进程多线程的架构）。这也就是说，MySQL数据库实例在系统上的表现就是一个进程。

在Linux操作系统中启动MySQL数据库实例，命令如下所示：

```
[root@xen-server bin]# ./mysqld_safe &
```

```
  [root@xen-server bin]# ps -ef | grep mysqld
  root      3441  3258  0 10:23 pts/3    00:00:00
/bin/sh ./mysqld_safe
mysql 3578 3441 0 10:23 pts/3 00:00:00
/usr/local/mysql/libexec/mysqld --basedir=/usr/local/mysql
--datadir=/usr/local/mysql/var --user=mysql
--log-error=/usr/local/mysql/var/xen-server.err
--pid-file=/usr/local/mysql/var/xen-server.pid
--socket=/tmp/mysql.sock --port=3306
  root      3616  3258  0 10:27 pts/3    00:00:00 grep mysqld
```

请注意进程号为3578的进程，该进程就是MySQL实例。这里我们使用了mysqld_safe命令来启动数据库，启动MySQL实例的方法还有很多，在各种平台下的方式可能会有所不同。在这里不一一举例。

当启动实例时，MySQL数据库会去读取配置文件，根据配置文件的参数来启动数据库实例。这与Oracle的参数文件（spfile）相似，不同的是，在Oracle中，如果没有参数文件，启动时会提示找不到该参数文件，数据库启动失败。而在MySQL数据库中，可以没有配置文件，在这种情况下，MySQL会按照编译时的默认参数设置启动实例。用以下命令可以查看，当MySQL数据库实例启动时，它会在哪些位置查找配置文件。

```
[root@xen-server bin]# ./mysql --help | grep my.cnf
order of preference, my.cnf, $MYSQL_TCP_PORT,
/etc/my.cnf /etc/mysql/my.cnf /usr/local/mysql/etc/my.cnf ~/.my.cnf
```

可以看到，MySQL是按/etc/my.cnf→/etc/mysql/my.cnf→/usr/local/mysql/etc/my.cnf→~/.my.cnf的顺序读取配置文件的。可能有人会问：“如果几个配置文件中都有同一个参数，MySQL以哪个配置文件为准？”答案很简单，MySQL会以读取到的最后一个配置文件中的参数为准。在Linux环境下，配置文件一般放在/etc/my.cnf下。在Windows平台下，配置文件的后缀名可以是.cnf，也可能是.ini。运行mysql -help命令，可以找到以下的内容：

```
Default options are read from the following files in the given order:
C:\Windows\my.ini C:\Windows\my.cnf C:\my.ini C:\my.cnf
C:\Program Files\MySQL\M\MySQL Server 5.1\my.cnf
```

配置文件中有一个datadir参数，该参数指定了数据库所在的路径。在Linux操作系统下，datadir默认为/usr/local/mysql/data。当然，你可以修改该参数，或者就使用该路径，但该路径只是一个链接，如下所示：

```
mysql> show variables like 'datadir'\G;
*************************** 1. row ***************************
Variable_name: datadir
        Value: /usr/local/mysql/data/
1 row in set (0.00 sec)1 row in set (0.00 sec)

mysql>system ls -lh /usr/local/mysql/data
total 32K
drwxr-xr-x  2 root mysql 4.0K Aug  6 16:23 bin
drwxr-xr-x  2 root mysql 4.0K Aug  6 16:23 docs
drwxr-xr-x  3 root mysql 4.0K Aug  6 16:04 include
drwxr-xr-x  3 root mysql 4.0K Aug  6 16:04 lib
drwxr-xr-x  2 root mysql 4.0K Aug  6 16:23 libexec
drwxr-xr-x 10 root mysql 4.0K Aug  6 16:23 mysql-test
drwxr-xr-x  5 root mysql 4.0K Aug  6 16:04 share
drwxr-xr-x  5 root mysql 4.0K Aug  6 16:23 sql-bench
lrwxrwxrwx  1 root mysql   16 Aug  6 16:05 data -> /opt/mysql_data/
```

从上面可以看到，其实data目录是一个链接，该链接指向了/opt/mysql_data目录。当然，必须保证/opt/mysql_data的用户和权限，使得只有mysql用户和组可以访问。

1.2 MySQL体系结构

由于工作的缘故，我很大一部分时间都花在对开发人员进行数据库方面的沟通和培训上。此外，不论他们是DBA，还是开发人员，似乎都对MySQL的体系结构了解得不够透彻。很多人喜欢把MySQL与他们以前使用过的SQL Server、Oracle、DB2作比较。因此，我常常会听到这样的疑问：

- 为什么MySQL不支持全文索引？
- MySQL速度快是因为它不支持事务？
- 数据量大于1 000W时，MySQL的性能会急剧下降吗？

……

对于MySQL的疑问还有很多很多，在我解释这些问题之前，我认为，不管使用哪种数据库，了解数据库的体系结构都是最为重要的。

在给出体系结构图之前，我想你应该理解了前一节提出的两个概念：数据库和数据库

实例。很多人会把这两个概念混淆，即MySQL是数据库，MySQL也是数据库实例。你这样来理解Oracle和SQL Server可能是正确的，但这对于以后理解MySQL体系结构中的存储引擎可能会带来问题。从概念上来说，数据库是文件的集合，是依照某种数据模型组织起来并存放于二级存储器中的数据集合；数据库实例是应用程序，是位于用户与操作系统之间的一层数据管理软件，用户对数据库数据的任何操作，包括数据库定义、数据查询、数据维护、数据库运行控制等，都是在数据库实例下进行的，应用程序只有通过数据库实例才能和数据库打交道。

如果这样讲解后你还是觉得不明白，那我再换一种更直白的方式来解释：数据库是由一个个文件组成（一般来说都是二进制的文件）的，如果要对这些文件执行诸如SELECT、INSERT、UPDATE和DETELE之类的操作，不能通过简单的操作文件来更改数据库的内容，需要通过数据库实例来完成对数据库的操作。所以，如果你把Oracle、SQL Server、MySQL简单地理解成数据库，可能是有失偏颇的，虽然在实际使用中我们并不会这么强调两者之间的区别。好了，在给出上述这些复杂枯燥的定义后，现在我们可以来看看MySQL数据库的体系结构了，如图1-1所示。

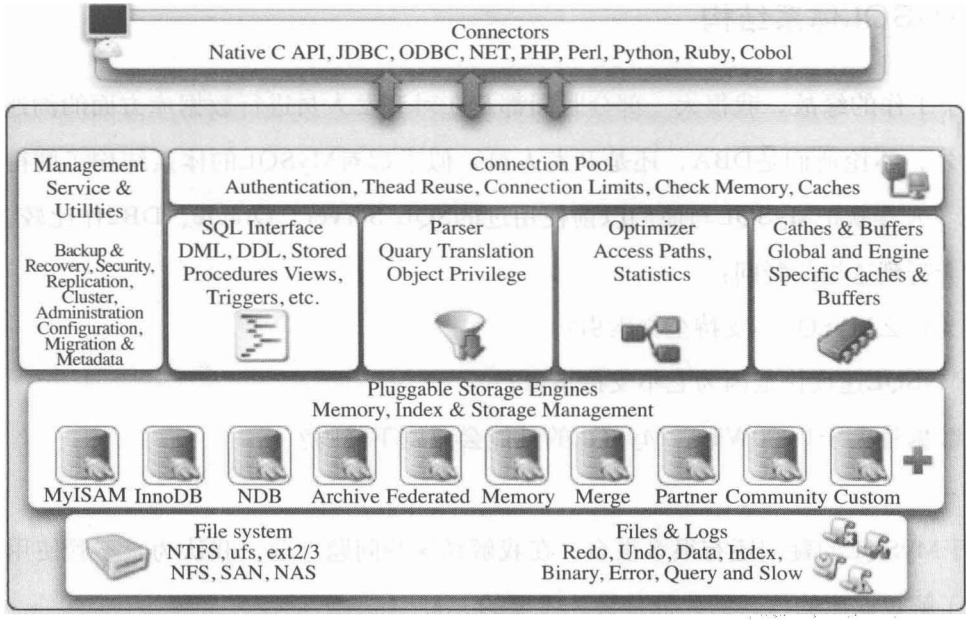

图1-1 MySQL体系结构

从图1-1我们可以发现，MySQL由以下几部分组成：

- 连接池组件。
- 管理服务和工具组件。
- SQL接口组件。
- 查询分析器组件。
- 优化器组件。
- 缓冲（Cache）组件。
- 插件式存储引擎。
- 物理文件。

从图1-1还可以看出，MySQL区别于其他数据库的最重要的特点就是其插件式的表存储引擎。MySQL插件式的存储引擎架构提供了一系列标准的管理和服务支持，这些标准与存储引擎本身无关，可能是每个数据库系统本身都必需的，如SQL分析器和优化器等，而存储引擎是底层物理结构的实现，每个存储引擎开发者都可以按照自己的意愿来进行开发。

注意：存储引擎是基于表的，而不是数据库。请牢牢记住图1-1所示的MySQL体系结构图，它对于你以后深入了解MySQL有极大的帮助。

1.3 MySQL表存储引擎

1.2节大致讲解了MySQL数据库独有的插件式体系结构，并介绍了存储引擎是MySQL区别于其他数据库的一个最重要特性。存储引擎的好处是，每个存储引擎都有各自的特点，能够根据具体的应用建立不同的存储引擎表。对于开发人员来说，存储引擎对其是透明的，但了解各种存储引擎的区别对于开发人员来说也是有好处的。对于DBA来说，他们应该深刻地认识到MySQL的核心是存储引擎。

由于MySQL是开源的，你可以根据MySQL预定义的存储引擎接口编写自己的存储引擎，或者是如果你对某种存储引擎不满意，可以通过修改源码来实现自己想要的特性，这就是开源的魅力所在。比如，eBay的Igor Chernyshev工程师对MySQL Memory存储引擎的