583

第 14 章

用于高性能 MySQL 的工具

Tools for High Performance

MySQL 服务器的发布包没有包含那些能完成许多常见任务的工具，例如监控服务器的工具、比较服务器间数据的工具。所幸的是，MySQL 志愿者社区已经开发了多种多样的工具，帮你解决各种问题。许多公司也提供了商业化的替代工具或是对 MySQL 已有工具的补充。

本章内容将遍及一些最常用最重要的产品化 MySQL 工具。我们把这些工具分成以下几类：界面、监控、分析和辅助工具。

14.1 带界面的工具

Interface Tools

带界面的工具可以帮你运行查询、创建表和用户、完成其他常见的任务。这一节里将简要地描述一下此类工具里最常用的几个。这些工具能做的事情，你用 SQL 查询和命令也是能做到的，但是使用它们会更加方便，即能避免错误，又能加快你的工作进度。

14.1.1 MySQL 可视化工具

MySQL Visual Tools

MySQL AB 发布了一套可视化工具，其中包括了 MySQL 查询浏览器（MySQL Query Browser）、MySQL 管理员（MySQL Administrator）、MySQL 迁移工具箱（MySQL Migration Toolkit）和 MySQL 工作台（MySQL Workbench）。这些工具都可以免费使用，你可以把它们一起下载下来安装在电脑上。它们能在所有流行的操作系统上运行，此前其中还带有一些恼人的小问题，但是，MySQL AB 已经找出并修复了这些 bug。

MySQL 查询浏览器可以用于运行查询、创建表和存储过程、导出数据和浏览数据库结构，它的使用帮助已经集成在 MySQL 的 SQL 命令和函数手册里。对于那些开发和查询 MySQL 数据库的人来说，MySQL 查询浏览器是最有用的工具。

584

MySQL 管理员的功能集中在服务器管理上，所以，它最适合 DBA 使用，而不是开发人员和分析人员。它可以帮助 DBA 把创建备份、创建用户并分配权限、显示服务器日志和状态信息等过程进行自动化处理。它还包括了一些基本的监控功能，例如图形化的状态变量显示，但是它没有下文里会提到的交互式监控工具那么灵活，也无法为后续的分析工作记录当前的系统统计信息——这对其他许多监控工具来说是都能做到的。

这个工具集里还包括了 MySQL 迁移工具箱，它可以帮你把数据从别的数据库系统迁移到 MySQL 里。工具集里的 MySQL 工作台是一个建模工具。

MySQL 自带工具的好处是它们是免费的，而且质量也相当不错，能够在许多桌面操作系统上运行。它们有一套简洁的功能特性，可以胜任许多任务。其中最突出的功能就是 MySQL 管理员里的用户管理和备份，以及 MySQL 查询浏览器里的集成文档。

这些工具的主要缺点是稍微有点简单，如果少一些华而不实的功能，可能会更合乎高级用户的需求。关于这个工具集的完整的描述，包括界面截图，你可以在 MySQL 的网站 *http://www.mysql.com/products/tools/*上找到。

提示：MySQL 工作台最近已经被从头改写了一遍，并推出了免费的和商用的两个版本。免费版的功能并没有缩水，但是商业版里包含了一些插件，可用于一些任务的自动化，减少了手工操作。在写下这段话的时候，MySQL 工作台的版本仍然是 beta 版。

14.1.2 SQLyog

SQLyog 是最常用的 MySQL 可视化工具，它设计良好，专门支持 DBA 和开发人员的工作。罗列它全部功能特性的清单会很长，所以就不包括进来了，下面只提一下其中的“亮点”：

- 代码自动完成能帮你更快地写查询语句。
- 能够通过 SSH 隧道连接到远程主机服务器。
- 可视化工具和向导可以帮你像构建查询语句一样完成普通的任务。
- 可以进行任务调度，在预定时间里执行备份、数据导入和数据同步等操作。
- 有键盘快捷键功能。
- 架构比较，它提供了访问对象（例如表、视图）属性的功能。
- 用户管理。

SQLyog 拥有你所期望的所有标准功能，例如配置编辑器。这个工具只能在 Microsoft 的 Windows 平台上使用，全功能版的需要出钱购买，有限功能版是免费的。更多关于 SQLyog 的信息可以访问 *http://www.Webyog.com*。

585

14.1.3 phpMyAdmin

phpMyAdmin 是一款很流行的管理工具，它在一个 Web 服务器上运行，让你可以通过基于浏览器的界面来管理你的 MySQL 服务器。在查询和管理方面，它有着许多非常出彩的功能特性。它主要的优点是平台独立性、一个庞大的功能集，以及可以通过浏览器访问。当你远离服务器环境并且只有一个浏览器可供你使用时，基于浏览器访问的功能会是非常有用的。举例来说，你可以在一台只拥有 FTP 访问权的主机服务器上安装上 phpMyAdmin，而原本连 mysql 客户端或其他 shell 程序都无法运行。

phpMyAdmin 是个很轻便的工具，在各种情况下都能找出满足你所需要的功能。当你在一个可以提供 Web 访问的系统上安装 phpMyAdmin 的时候一定要小心，因为如果你的服务器安全无法保证，那就是给攻击者提供了一条便捷的入侵路径。

但是，也有 phpMyAdmin 的反对者们声称 phpMyAdmin 的功能太多太庞大了，错综复杂。phpMyAdmin 常驻在 SourceForge.net 上，在那里它一直是最优秀的项目之一。更多信息可以访问 *http://sourceforge.net/projects/phpmyadmin/*。

14.2 监控工具

MySQL 的监控可以作为一个独立的主题写成一本书：它是一个很大很复杂的任务，不同的应用有着不同的需求。但是，我们可以针对这个标题把一些很好的工具和资源介绍给你。

“监控”是被大家滥用的术语之一，承载了几重意思。但是，在我们的经验里，多数基于 MySQL 的购物网站需要做许多不同类型的监控任务。

我们讲到的监控工具被分为非互动的和互动的两类。非互动监控常常就是一个自动化系统，它接收系统的测量值，如果有超出安全范围的，就通过发出警告提醒管理员。互动监控工具可以让你实时地观测服务器。在下面几个小节里，我们会把这两类工具分开来讲。

你可能对监控工具在其他方面的差别也很感兴趣，例如被动监控和主动监控，后者会发送警报信息并作出初步反应（就像 Nagios 一样）；或者你可能正在寻找这样一个工具：它可以创建一个信息仓库，而不仅仅是显示当前的统计信息。在接下来的介绍里，我们将会逐个指出每一种工具的特点。

14.2.1 非交互式监控系统

586

有许多监控系统不是专为监控 MySQL 而设计的，它们就是一个通用系统，里面设计了一个周期性的任务，定时去检查各类资源的状态，例如像服务器、路由器、以及各种软件（包括 MySQL）。它们常常会提供一个插件架构，同时有一个预定的 MySQL 插件可供使用。这样的一些系统能够记录监控对象的状态，并通过 Web 界面用图形化的形式表示出来。当监控对象出现问题，或者状态值超过安全范围时，它们还能发送报警信息，或者执行一个初始化的动作。

一般而言，你要把这个监控系统安装在自己的服务器上，通过它去监控其他服务器。如果你正在用它监控着一台重要的服务器，那么它很快就会成为你的 IT 基础架构里关键的一部分，因此，你还要采取一些额外的步骤，例如监控系统本身也必须是冗余的，以便于故障恢复。

一个自动化的监控系统能记录历史信息，当一个 MySQL 实例因为负载增加或者遇到其他故障时，监控系统显示出的状态变化趋势会成为“救命恩人”。修复问题常常需要知道哪些信息发生了变化，这就要求了解你的服务器的历史信息。当监控系统发现有些信息不对头时，它会在灾难发生前及时警告你，帮你集中精力发现并修复这个即将发生的故障。

自己开发的系统

许多组织在开始时都是自己构建监控报警系统。当只有几个系统需要监控，也没几个人使用时，它们往往都能工作得很好。然而，当组织逐渐扩大，变得更复杂，越来越多的系统管理员加入进来后，自产的监控系统就会趋于崩溃。每次当网络发生故障时，它就会发送出数以千记的消息把邮箱塞满；或者当有关键性问题发生时，它就悄悄地崩溃了，也不提醒任何人。对于自产的监控系统，重复或冗余的通知信息是最常见的问题，会妨碍到正常工作的完成。

如果你正打算自己编写一个监控工具——哪怕只要求像 cron 那么简单的功能：检查一个查询，如有问题就用邮

件通知某人，那么，你应该仔细考虑下面这些建议。最好还是把时间和精力花在学习使用下文中要提到的监控系统之一上。即使这些系统里的某几个有着陡峭的学习曲线，看起来不值得初始投资，但是，从长远来看，它们可以帮你节省时间和精力，使你组织里的系统状态渐入佳境。使用它们中的一个，即使开始时会做得很糟糕，但最终会被证明这是比实现自产系统更合适的选择。至少从长远来看，你在使用一个标准的监控系统时将获得经验和技能。

Nagios

587

Nagios（*http://www.nagios.org*）是一个开源的监控报警系统，它会定时检测你定义的服务，并将检测结果跟默认值或者上下限作比较。如果结果值超出了范围，Nagios 就会执行一段程序并且/或者把这个问题提醒某人。Nagios 的联系人和报警系统可以让你把通知发给不同的联系人。根据日期时间或其他条件，你也可以更改警报，或者把警报发送到其他地方，或者预告停机时间。Nagios 也理解服务之间的依赖关系，因此，当 MySQL 实例无法启动时，如果原因是网络中间的路由器故障或主机服务器崩溃而使服务器不可到达的话，Nagios 就不会来打扰你。

Nagios 可以将任何可执行文件作为插件来运行，只要给予它正确的参数，就能输出正确的结果。因此，Nagios 的插件可以是各种语言编写的，包括 shell 语言、Perl、Ruby 和其他脚本语言。*http://www.nagiosexchange.org* 站点就是 Nagios 专门的共享插件和分类站点。如果你在上面找不到你所需要的插件，自己创建一个也很简单：只要能接受标准的参数，并有正确的退出状态就可以，你也可以有选择地加入一些能让 Nagios 捕获的输出信息。

在许多操作系统上，Nagios 通过几种方法（包括主动检查、远程执行插件和被动检查——主要是接收其他系统“推”过来的状态数据），几乎可以监控你能测量到的一切状态。它还带有一个 Web 界面，通过这个界面，你可以检查状态、显示图表、可视化你的网络和当前的状态、安排计划中的停机时间等等。

Nagios 主要的缺点在于它让人望而生畏的复杂性。哪怕你一时学得很好，但到用的时候还是有点难度。它还把所有的配置项目都放在文件里，每个配置项目的语法都很特别，很容易出错。当你的系统日渐庞大的时候，更改这些配置信息会是一件艰苦的体力活。最后，它的图表、趋势图和可视化功能都相当有限。Nagios 把一些性能信息和其他数据存储在一个 MySQL 服务器上，然后可以基于这些数据来生成图表，但是，对于其他系统，兼容性还是不够的。

有几本专门讲述Nagios的书，我们比较喜欢Wolfgang Barth的《Nagios System and Network Monitoring》(No Starch Press)。

可替代 Nagios 的工具

虽然 Nagios 是最常用的多功能监控报警软件（注 1），但是，还是有其他几款开源工具可供你选用：

Zenoss

588

> Zenoss 是用 Python 编写的，拥有基于浏览器的用户界面，并使用了 Aajax 使操作更加快捷而富有效率。它将监控、报警、趋势显示、图表显示和记录历史数据等功能合成在一个统一的工具里，它还能在网上自动发现资源，在默认情况下，Zenoss 使用 SNMP 从远程机器上收集数据，但它也可以使用 SSH，并且支持 Nagios 插件。想要了解更多相关信息，可以访问 *http://www.zenoss.com*。

注 1：可能就是因为一旦你安装、配置好 Nagios，你就永远不会再想到监控系统了。

Hyperic HQ

Hyperic HQ 是一款基于 Java 的监控系统，它的目标跟其他同类别的软件不太一样，它要成为企业级的监控系统。跟 Zenoss 一样，它也能自动发现资源，支持 Nagios 插件，但是它的逻辑组织和架构很不一样，显得有一点庞大。至于它是不是适合你的需要，那要看你的参数设置和监控的方式了。更多关于 Hyperic HQ 的信息，可以访问 *http://www.hyperic.com*。

OpenNMS

OpenNMS 是由 Java 编写的，拥有一个活跃的开发者社区。它具备了常规的功能，例如监控和报警，也加入了图表和趋势显示的功能。它的目标是高性能、伸缩性、自动化以及良好的兼容性。跟 Hyperic 一样，它也企图成为一款企业级的监控软件，可以用在大型的关键系统上。更多关于 OpenNMS 的信息，可以访问 *http://www.opennms. org*。

Groundwork Open Source

Groundwork Open Source 实际上是基于 Nagios 的，它把 Nagios 和其他几个工具集成为一个系统，并安上一个统一的门户界面。描述它的最好方法可能就是：如果你对 Nagios、Cacti 及其他工具很熟悉，并且能够花大量的时间把它们无缝地集成在一起的话，你也能在家庭作坊里做一个出来。关于它的更多信息，可以访问 *http://www.groundwor- kopensource.com*。

Zabbix

Zabbix 是一个开源监控系统，在许多方面跟 Nagios 很相像，但是也有一些关键的不同点。例如，它把所有配置信息和其他数据都存放在一个数据库里，而不是放在配置文件里；它比 Nagios 存储了更多类型的数据，这样可以生成更好的趋势图和历史报告。它的网络图表和可视化功能也优于 Nagios。很多使用它的人发现它更易于配置，更具有兼容性。说起来它也能比 Nagios 承受更大的负载。在另一方面，Zabbix 的社区要比 Nagios 少，它的报警功能也不够高级。更多关于 Zabbix 的信息，可以访问 *http://www.zabbix.com*。

MySQL 监控和建议服务

MySQL 自己的监控方案就是设计用来监控 MySQL 实例的，但也能够监控主机的一些关键方面。这个工具不是开源的，需要 MySQL 企业订阅费。

589

与 Nagios 相比，这个服务的主要优点在于它提供了一套预设的规则或者说是“顾问”，通过这些规则，它会去检查服务器性能、状态和配置等许多方面。当它注意到有问题发生时，它会向用户建议解决方案，而不仅仅是向管理员报告说发生了什么错误。它也有一个设计良好的仪表板式界面，能够即刻显示所有服务器的状态信息。

尽管采用 Nagios 或其他系统来监控可能也可以获得到同样的统计数据，但是，对于 Nagios 来说，要想获取到 MySQL 里大量的测量值，它还要增加许多工作量用于编写插件、修改配置之后才能做到。而对于 MySQL 监控和建议服务，这些都是唾手可得的。

这个产品也有一个缺点，就是你无法使用它去监控网络的其余部分，它只是用来监控 MySQL 的。而且，它还要在每一台被监控的服务器上安装一个代理，一些 MySQL 管理员对此会很反感，他们希望服务器除了核心服务外，越简单越好。

更多信息，可以访问 *http://www.mysql.com/products/enterprise/advisors.html*。

MONyog

MONyog（*http://www.Webyog.com*）是一个轻量级的无代理的监控系统，它跟以上那些工具有着不同的实现方

法：它的底层是一个 JavaSscript 引擎，所有配置都是通过 JavaScript 对象模型来完成的。它被设计为在桌面系统上运行，运行时它会在一个闲置的端口上打开一个 HTTP 监听器。这样，你就可以把你的浏览器指向这个端口，查看到 MySQL 服务器的信息了，这些信息都是结合了 Javascript 和 Flash 来表示的。

MONyog 实际上有交互式和非交互式两种类型，因此，你可以把这两种类型的监控功能都尝试着用用看。

基于 RRDTool 的系统

严格地说，RRDTool（*http://www.rrdtool.org*）不算是一个监控系统，但是，它很重要，有必要在此提一下。很多组织里都是使用几种脚本或程序——这些一般都是自制的——从服务器那里读取到信息，然后再保存到循环数据库（Round-robin database，RRD）文件里。在许多要获取记录生成图表的环境下，RRD 文件是一个很适合的解决方案。它们能自动聚合输入的数据，如果输入数据值没有按期在随后提交进来时，还能在随后插入这些丢失的数据。它们还都带有强大的图表工具，能够生成漂亮的与众不同的图表。现在已经有一些基于 RRDTOOL 的系统可供使用了。

Multi Router Traffic Grapher，或者叫 MRTG（*http://oss.oetiker.ch/mrtg/*）就是一款典型的基于 RRDTOOL 的系统。它真正的设计初衷是记录网络数据流，但是它也被扩展用来记录和图表化表示其他一些东西。

Munin（*http://munin.projects.linpro.no*）是一个能为你采集数据的系统，将它放入 RRDTool 后，就会根据数据生成不同粒度的图表。它能从配置信息里生成静态的 HTML 文件，这样你就可以轻松地浏览，查看趋势情况。要定义一个图表也很容易，只要你创建一个插入式脚本，它的命令行帮助输出结果需要有 Munin 能识别的特殊语法，Munin 把这些参数用作图表指令。Munin 的缺点包括它要求每个被监控的系统都要安装一个代理、简化了的“一个尺寸、大小通用”的配置方式和图表选项对于某些需要来说还不够有弹性。

Cacti（*http://www.cacti.net*）是另外一个常用的图表和趋势显示系统。它的工作方式是：从系统里获取数据，然后保存在 RRD 文件里，然后用 PHP Web 界面的形式，使用 RRDTool 把数据以图表的形式展示出来。这个显示界面也是配置和管理界面（配置信息存储在一个 MySQL 服务器里）。它是模板驱动的，因此，你可以自己定义模板，并放到你的系统里使用。它也可以从 SNMP 或者其他自定义脚本里获取到数据。

Cricket（*http://cricket.sourceforge.net*）是一个用 Perl 编写的跟 Cacti 类似的系统，使用的是基于文件的配置系统。Ganglia（*http://ganglia.sourceforge.net*）也跟 Cacti 类似，但它的设计初衷是用于监控集群和系统网格，因此，你可以查看到由许多服务器信息聚合得到的结果，也可以按照你的意愿，查看单独某台服务器的信息。（Cacti 和 Cricket 无法显示聚合数据。）

以上这些系统都可以被用作 MySQL 系统信息的收集、记录、图表化数据和报告，它们在用途方面差异较小，都具备了不同程度的兼容性。但是，它们缺乏真正意义上的兼容性，比如当某些东西出错时，它要能够有针对性地发送警报信息给某些人。它们中的一些甚至没有“错误”的概念。所以，有些人把这一点看作是此类系统的一大缺点，觉得最好还是把记录、图表化表示、报警这几项功能都独立开来。事实上，Munin 特地设计了使用 Nagios 来作为它的报警系统。然而，对于其他几个来说，这的确是缺点。另外还有一个缺点就是安装和配置这样一个系统，使其能完全满足你的需求，须投入很多时间和努力，不过，这一点也并不总会这样。

最后，你应该考虑到将来的需求。RRD 文件无法让你使用 SQL 或其他标准方法来查询它里面的数据。而且，在默认情况下，它永远会以一种恰好的粒度来存储数据，许多 MySQL 管理员就不愿意接受这种限制，转而选择一个关系数据库来存储这些历史数据。许多 DBA 也希望能以更个性化更有弹性的方式来记录数据，所以，他们转而编写他们自己的系统或修改一个现有的系统。

基于 RRDTool 的系统是不是正好能与你的组织相匹配，这还是要看个人的选择、管理系统所需的可用技能，以及你组织的具体需求。

14.2.2 交互式工具

交互式工具就是那些在你需要时就可以启动起来，并以视图显示的形式不断获取最新的服务器状态的软件。接下来，我们将主要讲解 innotop（*http://innotop.sourceforge.net*），但是，也有别的几个工具可供使用，例如 mtop（*http://mtop.innotop.sourceforge.net*）、mytop（*http://jeremy.zawodny.com/mysql/mytop/*）及一些基于 Web 的 mytop 克隆版本。

innotop

本书的作者之一——Baron Schwartz 编写了 innotop。别去理会它的名称，实际上它不仅仅用于监控 InnoDB 的内部信息。这个工具的设计灵感来自于 mytop，但是它提供了更多的功能。它有许多模式可用来监控所有类型的 MySQL 内部状态，包括 `SHOW INNODB STATUS` 里的全部可用信息，它能将这些信息分解成不同的部分。它也能让你同时监控多个 MySQL 实例，极具可配置性和可扩展性。

它的功能特性包括以下这些：

- 事务列表显示 InnoDB 当前的全部事务；
- 查询列表显示当前正在运行的查询；
- 显示当前锁和锁等待的列表；
- 服务器状态和变量的摘要信息显示了数值的相对变化幅度；
- 有多种模式可用来显示 InnoDB 内部信息，例如缓冲区、死锁、外键错误、I/O 活动情况、行操作、信号量，以及其他更多的内容；
- 复制监控，将主机和从服务器的状态显示在一起；
- 有一个显示任意服务器变量的模式；
- 服务器组可以帮你更方便地组织多台服务器；
- 在命令行脚本下可以使用非交互式模式。

innotop 很易于安装，你可以将它从操作系统的 package 库里取出来安装，也可以从 *http://innotop.sourceforge.net* 上下载到本地，然后解压缩，运行标准的 `make` 安装过程：

```
perl Makefile.PL
make install
```

一旦你安装完成，就可以在命令行里执行 innotop，它会带你完成连接到 MySQL 实例的过程。它会自动去读取你的~/.my.cnf 选项文件，这样，你除了输入服务器的主机名和按几次 Enter 键之外，什么都不用做。连接完成以后，就处在 T 模式（InnoDB Transcation）里了，这时，你应该看到 InnoDB 事务列表，如图 14-1 所示。

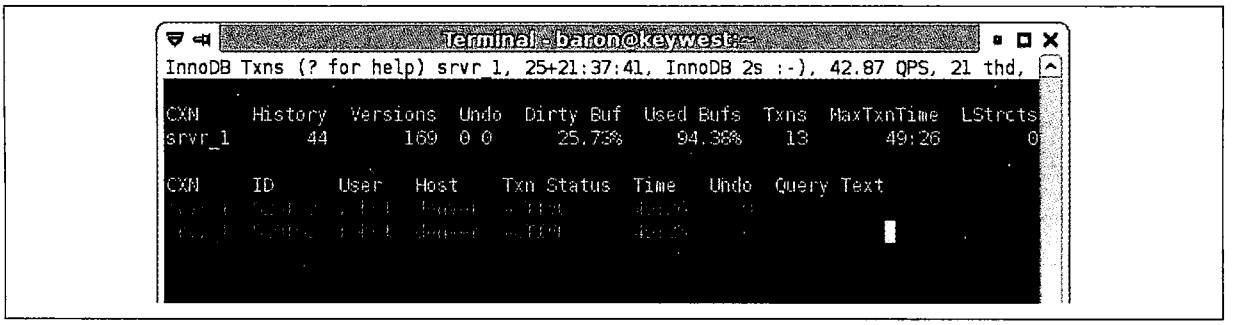

图 14-1：innotop 里的 T（Transaction）模式

默认情况下，innotop 采用过滤器来减少零乱的信息（就是 innotop 能显示出的所有信息，你可以定义自己的过滤器，或者定制它内部的过滤器）。在图 14-1 里，大多数事务都已经被过滤掉了，只显示出当前活动的事务。你可以按 i 键关闭过滤，让数量众多的事务信息把整个屏幕填满。

Innotop 在这个模式下会显示一个头部信息和一个主线程列表。头部信息里是一些 InnoDB 的总体信息，例如历史清单的长度、还未清除的 InnoDB 事务的数目、脏缓冲区在缓冲区池里所占的百分比等等。

这时你要按的第一个键应该是问号（?），查看一下帮助信息。虽然在屏幕上显示出的帮助内容会根据当前不同的模式而不同，但是，每一个活动的键总是会显示出来，这样，你就能看到所有可执行的动作了。图 14-2 显示的是 T 模式下的帮助信息。

我们在这里不会讲到所有的模式，但是，你从帮助信息里就能看出，innotop 有许许多多功能特性。

唯一要提及的是一些基本的自定义功能，告诉你如何监控你想要监控的信息。innotop 的强大功能之一就是能够解读用户定义的表达式，例如 `Uptime/Questions` 的意思是每秒钟的查询数，innotop 会显示自服务器启动以来和/或自上次采样之后累加起来的结果值。

于是，往显示表格里添加你自己的列就方便很多。举例来说，在 Q（Query List）模式下，有一个头部信息能显出服务器的一些总体信息。让我们看看怎么将它修改一下，使它能显示出索引键缓存有多满。启动 innotop，按下 Q 键进入 Q 模式。这时的操作结果看起来像图 14-3 一样。

这个屏幕截图已有过删减，因为在这个练习里，我们对查询列表没有兴趣。我们只关心头部信息。

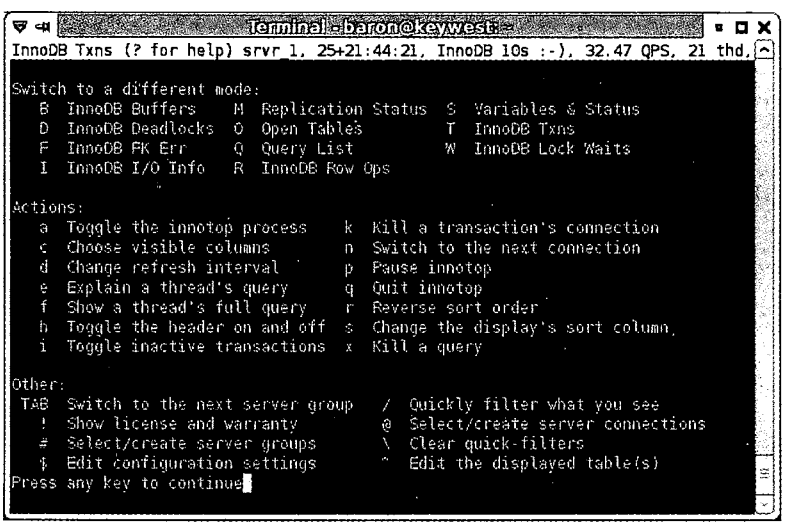

图 14-2：innotop 帮助信息

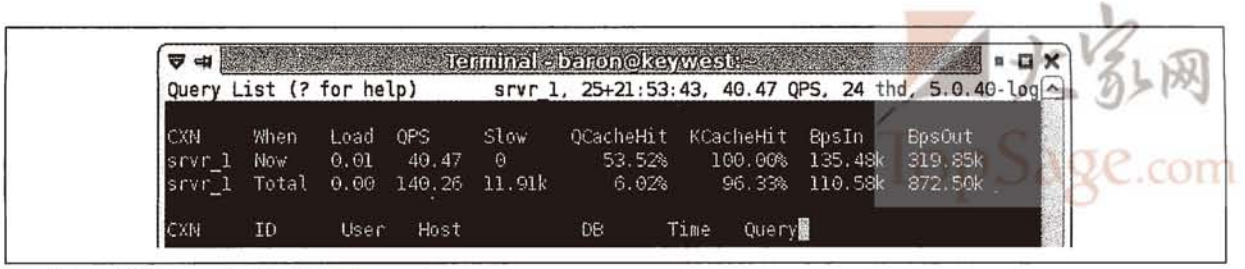

图 14-3：Q(Query List)模式时的 innotop

头部显示的是“现在”（它衡量的是自上次 innotop 从服务器取得新数据之后的增长情况）和“总共”（它衡量的是自 MySQL 启动后 25 天来的总体活动情况）的统计信息。头部里的每一列都是源自 SHOW STATUS 和 SHOW VARIABLES 里对应变量的值。像图 14-3 显示的头部是内建的，但是，要添加你自己的列也很容易。所有你要做的就是在头部“表格”里增加一列：按下^键打开表格编辑器，然后，在参数输入框里输入 q_header 开始编辑头部表格（图 14-4）。Tab 补全操作是内建的，因此，你可以按 q 然后再按 Tab 来完成整个词的输入。

在此之后，你将看到 Q 模式下头部表格的定义（图 14-5）。这个表格定义显示了表格的各个列。第 1 列是选择记号，我们可以移动这个选择记号，然后进行记录、编辑及其他操作。

图 14-4：增加一个头部信息（开始）

（按下?可以查看到整个列表），但是，此时我们只要创建一个新的列：只须按下 n 键，然后输入新列的名称（图 14-6）即可。

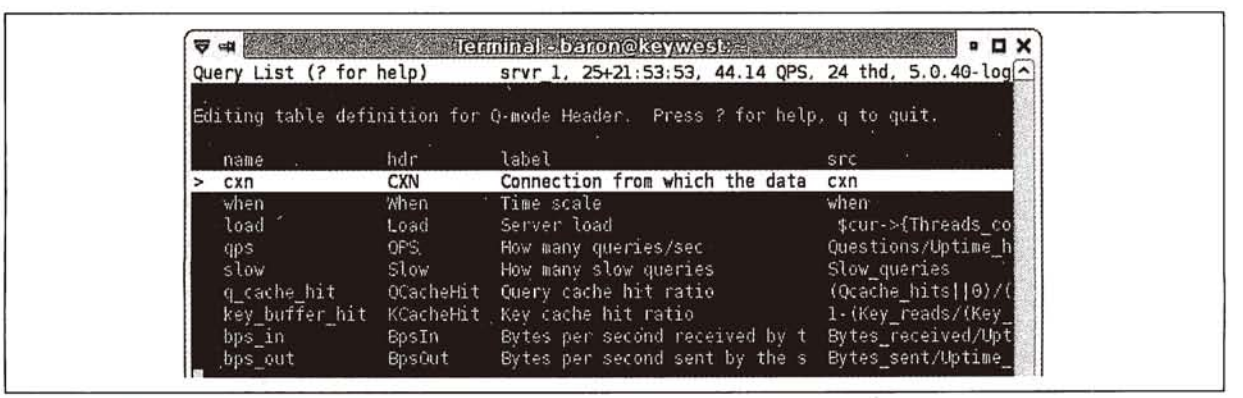

图 14-5：增加一个头部信息（选择）

图 14-6：增加一个头部信息（对列进行命名）

下一步就是输入列的头部名称，它会出现在列的顶端（图 14-7）。最后，选择列的来源，这是一个表达式，innotop 会将它编译成一个内部函数，你可以使用来自 SHOW VARIABLES 和 SHOW STATUS 的变量名。我们使用了一些括号和 Perl 式的“或”默认用来避免被零除，因此，等式看起来会比较直观。这里，我们也使用了一个名叫 percent() 的 innotop 转换函数，把结果值格式化为百分比的形式，更多的用法可以查看 innotop 文档。图 14-8 显示了的就是这样一个表达式。

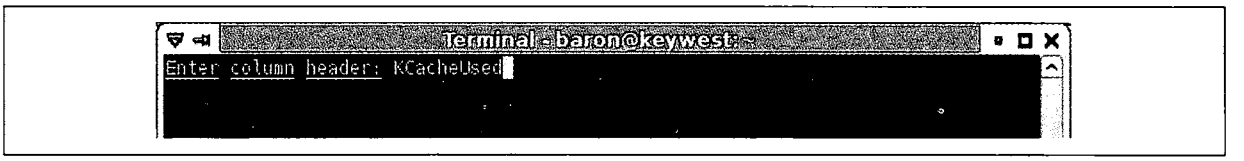

图 14-7：增加一列（列的文本标题）

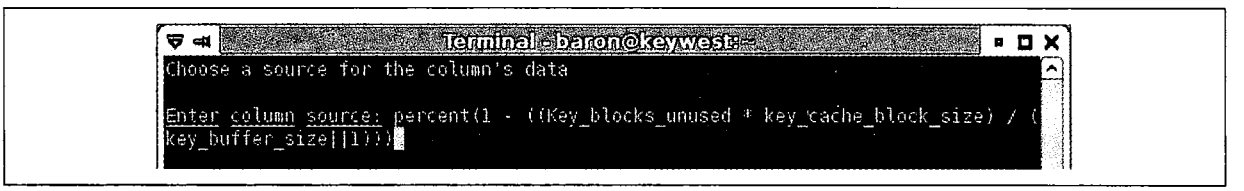

图 14-8：增加一列（要计算的表达式）

按 Enter，你就会看到表格的定义跟原先一样，但是在底部多了新增的一列。按住+键一会儿，把它从列表的底部移上去，到 key_buffer_hit 列的旁边，然后再按 q 退出表格编辑器。瞧：你新增的列就在 KcacheHit 和 BpsIn 之间（图 14-9）。要定制 innotop 监控你想要的东西也是非常容易的，即使有它做不到的事情，你也可以通过编写插件来实现。在 *http://innotop. sourceforge.net* 上有更多相关文档可供参考。

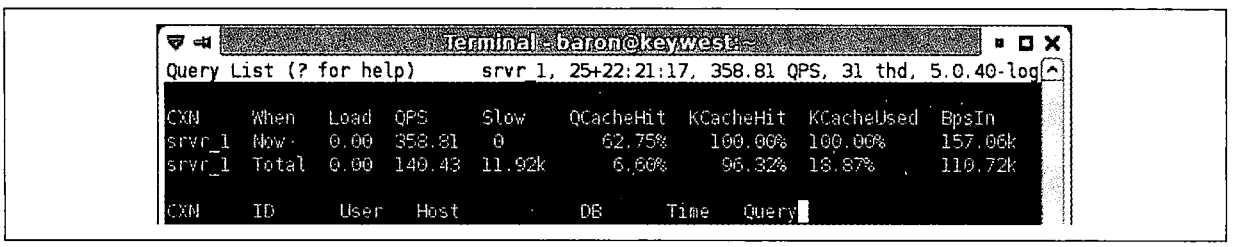

图 14-9：增加一列（最终效果）

14.3 分析工具

Analysis Tools

分析工具可以帮你自动化那些单调乏味的工作，例如监测服务器，找出还可以优化和调优的功能区域。这些工具可以作为解决性能问题的良好开端。如果它们中的一个发现了一个明显的问题，你就可以在这问题上全力以赴，也许能因此更快地解决它。

14.3.1 HackMySQL 工具

HackMySQL Tools

Daniel Nichter 维护着一个名叫 HackMySQL 的网站，在那里他提供了几款有用的 MySQL 工具。Mysqlreport 是一个 Perl 脚本，它能检查 SHOW STATUS 的输出结果，把它们转换为易读的报告，并打印出来。阅读这个报告要比你自己检查 SHOW STATUS 要快很多，而且信息更全面。

下面是这份报告里各主要部分的概述，以 3.23 版本为例：

- “Key” 段显示的是你的键（索引）的使用情况。如果这些值不太“健康”，你就可能需要调优一下你的索引键缓存的设置了。
- “Questions” 段显示的是服务器正在执行的是哪一类查询，它可以帮你找出系统的负载主要集中在何处。
- “SELECT and Sort” 段显示的是服务器最常执行的是哪几种查询计划和排序策略。通过这一节信息，你可以找出索引问题或者那些不够优化的查询。
- “Query Cache” 段显示了你的查询缓存的使用情况。如果它有问题存在，那你应该调优它的设置，否则，你的工作负载就无法利用缓存的好处了，甚至会停掉缓存。
- 还有一些段显示了表、锁、连接和网络数据流的信息。这里如果有问题，一般就是表明服务器还有需要调优的地方。
- 还有 3 个段显示的是 InnoDB 性能度量值和设置参数。此处的问题可能是源自不当的服务器设置、硬件故障、查询或式样缺少优化等原因。

在 *http://hackmysql.com/mysqlreport* 上有更多此类内容，包括一份怎么解读报告的详尽指导。特别是你如果要经常在不熟悉的服务器上排除故障，花一些时间学习报告的解读方法是很值得的。通过几次实践，你就能浏览一下报告就能立即找出问题的所在了。

Mysqlsla（MySQL Statement Log Analyzer）是另一款有用的工具。你可以用它分析由服务器上所有查询生成的日志、慢速查询日志（慢速的意思是该查询所需要的时间超出了配置里的最大时间）、或者任何其他日志。它可以接纳多种日志格式，然后能即刻对它们进行分析。更多关于分析 MySQL 日志的内容，请查看第 65 页的“更精细地控制日志”。

该网站上的其他程序可以帮你分析一台服务器的索引使用情况，以及检查跟 MySQL 相关的网络数据流。

14.3.2 Maatkit 分析工具

Maatkit 是 Baron Schwartz 的另一项创造，是一系列命令行工具的集合。所有工具都是用 Perl 编写的，用来补充 MySQL 未能提供的那些重要功能。你可以在 *http://maatkit.sourceforge.net* 上找到它们，包括了分析工具和其他一些小功能。

这些分析工具里的其中一个是 mk-query-profiler，当它监测你的服务器状态变量时能够执行查询，并会打印出一个具体且易读的关于查询执行前后的系统状态变量差异情况的报告。这份报告能使你对你的查询能有更深的理解，而不仅限于在它执行的时候。

你可以将查询通过管道传入 mk-query-profiler 的标准输入口，指定一个或多个查询文件，或者仅仅是要求它监测你的服务器而不运行任何查询（当你在运行一个外部应用时，这种方式会用得上）。你也能让它运行 shell 命令而不是查询。

mk-query-profiler 的报告分成好几个段落。默认的情况下，这个报告打印出的是一批摘要信息，但是，你也可以得到一份包含了每一个查询或者所选择的查询的报告，然后加载到 mk-profile-compact 的辅助工具里方便地进行比较。

以下是这份报告的主要段落：

- “Overall stats”段罗列了一些基本信息，例如执行时间、命令的数目和网络流量。
- “Table and index accesses”段显示了这个批处理里有多少个不同类型的执行计划。如果你看到有很多表扫描，这可能意味着你没有合适的索引可用于查询。
- “Row operations”段显示了底层处理器的数目和/或这个批处理产生的 InnoDB 操作的数目。糟糕的查询计划会引发更多的底层操作。
- “I/O operations”段显示的是这个批处理引发了多少内存和磁盘数据流。有一个可比较的段是显示跟 InnoDB 相关的数据操作。

总而言之，这份报告可以细致地描述出服务器正在处理多少种类型的任务，这比仅仅是衡量查询所需要的时间要有价值得多。例如，它可以帮你筛选这样两个查询：它们在低负载时，在一个小数据集上都用几乎相等的时间完成了查询；但是，当使用到的数据量变得很大的时候，或者说在高负载的时候，它们的表现就有了巨大的差异。它也可以被用来验证是否你的优化起到了作用，在这种应用环境下，它就像是一个微型的基准测试工具。

在这个工具箱里还有另外几种分析工具：

mk-visual-explain

重构从 EXPLAIN 里取得的查询执行计划，然后以更易读懂的树状图来显示。当查询计划很复杂的时候，这个尤为有用。查看 EXPLAIN 的输入，我们看到的是数百行语句，这样的长度几乎没人能理解它们。mk-visual-explain 作为一个教学工具也很有用，或者当你试着去学习怎么看懂 EXPLAIN 输出的时候，它也能帮得上忙。

mk-duplicate-key-checker

它能识别出重复的或冗余的索引和外键——这会影响到性能。更多内容请查看第 127 页的“多余和重复索引”。

mk-deadlock-logger

监测 InnoDB 里的死锁并将它们记录在一个文件或表里。

mk-heartbeat

精确测量复制的延迟时间，无须检查 SHOW SLAVE STATUS（它也不总是正确）。默认情况下，它保留了过去 1 分钟、5 分钟和 15 分钟里的移动平均值。这是针对本书第一版里那个心跳（Heartbeat）脚本提到过的更完整更具可配置性的实现。

14.4 MySQL 的辅助工具

MySQL 里有几个工具是为了消除 MySQL 提供的功能与它自带的命令行工具之间的隔阂。这一节里将讨论这些辅助工具。

14.4.1 MySQL Proxy

MySQL Proxy 项目是由 MySQL AB 开发并维护的，它以 GPL 许可证的形式发布，将来可能会随着 MySQL 服务器一起发布。在本书编写时，它诞生还不到一年时间，处于一个快速发展期里。（注 2）现在，你能在 *http://www.mysql. com* 的 Community 版块上找到它。它的文档已经集成在了 MySQL 使用手册里。

该软件的核心概念是一个带有状态的应用，它能解析 MySQL 的客户机/服务器协议，也能处于客户机与服务器之间，透明地转发它们的消息。一个客户端应用能够连接到它上面，跟连到服务器一样，这时，代理就会创建一个连接，连到一台真正的 MySQL 服务器——这种行为就像一个中间商。

这样一个单独的功能也可以被用于许多应用里（比如负载平衡和故障转移），但是代理更进了一步。既然它能解析客户端/服务器协议，那么它就能审查查询和响应结果。它还内建了一个 Lua 解释器，这样你就能编写自定义脚本，对查询和响应结果做任何你能想象到的操作，以下就是几种可能的应用方式：

- 重写或过滤查询。例如，你可以编写一个脚本识别出专门传递给代理的命令，然后做这些特定的操作，而不是把查询传递给服务器。
- 产生新的结果集，使它看上去像是来自 MySQL 服务器的，或者丢弃服务器生成的结果集。
- 基于它监测的内容，动态地调优 MySQL 服务器。例如代理能够开启或关闭那些缓慢的查询，或者能够跟踪查询统计信息，在响应一个查询时，显示出响应时间的柱状图。
- 在每次事务提交的时候注入查询语句（例如创建一个全局的事务标识）。

以上这些应用都有可工作的代码，你可以从在线文章和代码库里下载到。应用的可能性几乎是无限的。有创造力的用户会从中发掘出我们从没想到过的使用方法。如果你在考虑如何使用代理的时候有一些麻烦，我们建议你去阅读几篇 Giuseppe Maxia 或 Jan Kneschke 写的文章。

14.4.2 Dormando 的 MySQL 代理

另一个使用 GPL 授权的代理项目是 Dormando 的 MySQL 代理，它几乎跟 MySQL Proxy 同时出现（事实上是它最早引入 Lua 脚本功能）。它是对 MySQL Proxy 项目的一个响应，到目前为止还没有发布，而且最终的许可证方式也没确定下来。就像 MySQL Proxy 一样，它的改变也很快，因此，你应该去检查一下它最新的版本，看看目前处于什么样的状态。它的网站是 *http://www. consoleninja.net/code/dpm/*。

14.4.3 Maatkit 辅助脚本

我们在前面罗列分析工具的时候已经提到过 Maatkit 了，其实，它还包括了许多辅助脚本。在这些脚本里，最重要的是 mk-table-checksum 和 mk- table-sync，在第 380 页的“确定主、从服务器是否一致”里，我们已经讲到过它们。除此之外，Maatkit 还包括了以下这些脚本：

注 2：MySQL Proxy 进展迅速，当你在阅读本书的时候，这些信息可能已经过时了。

mk-archiver

运行清除和归档任务，帮你清理表里不需要的数据。这个工具是被设计用来移动数据的，而无需动用 OLTP 查询。但是，你也可以用它来建立数据仓库，或者查找和移除陈旧的数据。它能把数据写到一个文件里，并且/或者写到任何 MySQL 实例上的另一个表里。它还有一个插件机制，使定制任务更加容易实现，举例来说，你可以使用一个插件，让它在系统往日志表里插入数据时，自动在数据仓库里生成一个摘要表。

mk-find

跟 Unix 的 find 命令类似，但是，它是用于 MySQL 的数据库和表。

mk-parallel-dump

执行多线程逻辑备份。为了能在多 CPU 或多磁盘的系统里更快速地备份，它能把每个表分割成所需要大小的几个分块。实际上，你可以用它把任何工具都封装成多线程的形式，因此，它在多线程 `CHECK TABLE` 或 `OPTIMIZE TABLE` 操作方面也很有用处（以此为例）。许多类型的任务都能因此在多 CPU 或多磁盘获取并行处理的好处。

mk-parallel-restore

它是 mk-parallel-dump 的伴随程序。它并行地把文件加载到 MySQL 里。这个工具能够通过 `LOAD DATA INFILE` 直接加载分界符文件，或者将 SQL 文件委托给 mysql 客户端程序导入。它对许多加载操作做了一个灵活的封装，例如可以通过命名管道加载压缩文件。

mk-show-grants

能对 `GRANT` 语句进行规整化、排除、分离和排序，使它更易于在命令行里操作。有一种很有趣的应用是把你的数据库权限存储在一个版本控制系统里，不会有任何伪造的更改。

mk-slave-delay

使从服务器的更新滞后于它的主机，以便于灾难还原。如果有一条破坏性 SQL 语句在主机上执行后，你能在从服务器执行这条语句之前将它停止掉，重放二进制日志直到这条语句，然后将其提升为主机。这样的做法要快于在主机上重加载最近的一个备份，然后重新自备份以来的所有日志。

mk-slave-prefetch

实现了在第 402 页“为从服务器线程准备好缓存”里讨论过的技术。在某些工作负载下，它能使复制在从服务器上运行得更快。

mk-slave-restart

使从服务器在遇到一个错误后重新启动。

mk-table-checksum

在一台或多台服务器上并行地生成表内容的校验和，或者在各个复制里进行校验和查询，以检验你的从服务器上的数据是否一致。

mk-table-sync

更高效地找出表之间的差异，并生成用于解析它们的最小的 SQL 命令集。它也能在复制里面操作。

Baron 会经常加入新的工具，所以，这个列表可能也过时了。你可以在 *http://maatkit. sourceforge.net* 上获取到最新的工具和文档。

14.5 更多的信息来源

如果你正在 MySQL 上做着许多重复的或易出错的手工操作，而其他人可能已经创建了一个工具或脚本能减轻你的负担，问题在于你如何能找到这个工具。我们是通过阅读 Planet MySQL 博客聚合器（*http://www.planetmysql.org*）和 MySQL Forge 社区网站（*http://forge.mysql.com*）了解到许多我们喜爱的工具的。通常，这些网站都是学习 MySQL 的优秀资源。它们也有邮件列表、IRC 频道和论坛，在那里，你经常可以从那些友好的高手那里得到问题的解答（但是你要先搜索一下存档!）。

讨论会是另一个了解到 MySQL 的工具和技术的重要场所。即使你无法参加这样的讨论会，你也能经常性地下载到讲座的幻灯片或者在线观看会议的视频。

关于有些复杂工具（例如 Nagios）的更多信息，你也能找到专门讲解它们的书来看，在那些书里获得的资源将远远超过我们在本章里所提到的一点点相关内容。