

PROGRAMMER TO PROGRAMMER™



Professional SQL Server 2005 Performance Tuning

SQL Server 2005 性能调优

(美) Steven Wort 等著
Christian Bolton
齐宁 董泽惠 译



清华大学出版社

SQL Server 2005 性能调优

(美) Steven Wort 等著
Christian Bolton
齐宁 董泽惠 译

清华大学出版社

北 京

Steven Wort, Christian Bolton, et al.
Professional SQL Server 2005 Performance Tuning
EISBN: 978-0-470-17639-9
Copyright © 2008 by Wiley Publishing, Inc.
All Rights Reserved. This translation published under license.

本书中文简体字版由 Wiley Publishing, Inc. 授权清华大学出版社出版。未经出版者书面许可, 不得以任何方式复制或抄袭本书内容。

北京市版权局著作权合同登记号 图字: 01-2009-2061

本书封面贴有 Wiley 公司防伪标签, 无标签者不得销售。
版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

SQL Server 2005 性能调优/(美)沃特(Wort, S.), 波尔顿(Bolton, C.) 等著; 齐宁, 董泽惠 译。

—北京: 清华大学出版社, 2009.5

书名原文: Professional SQL Server 2005 Performance Tuning

ISBN 978-7-302-20087-1

I.S… II.①沃… ②波… ③齐… ④董… III.关系数据库—数据库管理系统, SQL Server 2005—教材
IV.TP311.138

中国版本图书馆 CIP 数据核字(2009)第 065516 号

责任编辑: 王 军 郑雪梅

装帧设计: 孔祥丰

责任校对: 胡雁翎

责任印制: 王秀菊

出版发行: 清华大学出版社

地 址: 北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编: 100084

社 总 机: 010-62770175

邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者: 清华大学印刷厂

装 订 者: 三河市金元印装有限公司

经 销: 全国新华书店

开 本: 185×260 印 张: 33 字 数: 803 千字

版 次: 2009 年 5 月第 1 版 印 次: 2009 年 5 月第 1 次印刷

印 数: 1~4000

定 价: 68.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题, 请与清华大学出版社出版部联系调换。联系电话: (010)62770177 转 3103 产品编号: 027544-01

前言

SQL Server 是一个非常成功的数据库服务器，它在自调校(self-tuning)方面非同一般。即装即用的 SQL Server 做了大量工作以良好地运行，而且可以在完全无须用户配置的条件下提供极佳的性能。随着廉价硬件的出现和数据的爆炸性增长，当前的 SQL Server 系统经常会运行在过去被企业级主机系统所控制的场景中。

随着廉价磁盘、磁盘控制器和内存的出现，几乎任何人都可以在一个不太大的系统上构建多个 T 字节的数据库。数据库规模上的大幅膨胀意味着更多的 SQL Server 系统正在冲击 SQL Server 自调校能力的边界。因此，很多 SQL Server 用户正在经历性能问题。

本书为所有咨询师、开发人员、数据库管理员以及任何必须第一次处理 SQL Server 性能问题的人提供了全面的资源。还适用于那些已经处理过 SQL Server 性能问题，但需要对如何进行性能调校进行新的审视的人。

本书从全新的视角来实现性能调校。书中演示了如何找到性能问题，而不是假定您已经知道问题是什么。

适合阅读本书的人员

如果您是咨询师、开发人员、DBA、架构师或对 SQL 性能感兴趣的人，那么本书适合于您。

阅读本书需要具有 T-SQL 的应用知识，而且知道如何执行基本的 SQL Server 和操作系统管理任务。有了这些基本知识，就可以开始对 SQL Server 进行性能调校了。

如果您是一名已经有一两年 SQL Server 开发经验的 SQL Server 开发人员，而且初次面临性能问题，那么您可以开始阅读本书。

如果您是一名 SQL Server 咨询师，而且正在去为客户解决性能问题的途中，那么您应当将本书放置到随身行李中，并且在到达客户所在地点的途中阅读。

如果您是一名 SQL Server DBA，而且您的系统的速度正在开始慢下来，那么这是一本非常适合您阅读的书。

本书的内容

本书包含了对 SQL Server 2005 进行性能调校的内容。

书中的内容是使用 SQL Server 2005 中可用特性编写的，有些例子使用了 SQL Server

2005 的新特性。这种情况下，例子不能在较早版本中运行。讨论的概念通常与任何 SQL Server 版本都相关。一些特定的、无法适用于较早版本的例子是基于新的特性的，如新的动态管理视图。

本书的结构

本书分为 4 个部分：

- 第 I 部分介绍如何找出是什么引起了性能瓶颈。
- 第 II 部分说明了如何去除瓶颈。
- 第 III 部分讨论了预防措施、基线以及各种有用的工具。
- 第 IV 部分讨论了关于如何从一开始就为良好的性能进行设计的话题，而不是采用首先创建功能设计，然后在设计完毕之后再考虑性能的方法。

第 I 部分 在问题发生时寻找瓶颈

第 1~5 章处理如何找到当前瓶颈。这是当您接到发生性能问题的用户打来的电话时应当查看的部分：

- 第 1 章介绍性能调校方法的总的方法学。
- 第 2 章讨论如何检查服务器资源，以了解是否存在服务器资源瓶颈。
- 第 3 章讨论如何监视 SQL Server 资源，以了解是否存在资源瓶颈。
- 第 4 章演示了如何使用 SQL Server 等待类型找出资源瓶颈以及引起这些瓶颈的查询。
- 第 5 章说明了如何使用 SQL Profiler 寻找长时间运行的查询。

在阅读完这一部分之后，应当能够找到一个瓶颈。下一步是寻找途径来去除瓶颈。这些内容包含在第 II 部分中。

第 II 部分 通过调校去除瓶颈

第 6~9 章介绍了可用来去除可能导致性能问题的各种类型的瓶颈的方法。一旦确定了一个瓶颈，通过学习这一部分，可以找出去除该瓶颈的方法。

- 第 6 章介绍了如何通过调校服务器配置的设置去除第 2 章中发现的瓶颈。
- 第 7 章阐述了如何通过调校 SQL Server 设置去除第 3 章中发现的瓶颈。
- 第 8 章讨论了模式的调校。如果发现基本模式中存在瓶颈，则可以查看本章内容来去除瓶颈。
- 第 9 章介绍了对 T-SQL 的调校，并演示了如何去除在第 4 章和第 5 章中发现的瓶颈。

在学习完这一部分之后，应当能够去除瓶颈。但是如何才能够在瓶颈发生之前就阻止它们呢？这将在第 III 部分中进行介绍。

第III部分 预防措施以及使用工具获得性能基线

这一部分介绍了应当在客户开始抱怨性能糟糕之前做的工作。通过采取这些预防措施，可以试着避免性能问题的出现。该部分介绍了使用不同工具来捕获关于系统基线性能的量度。

- 第 10 章介绍了使用 SQL Profiler 来重放工作负荷。这对创建性能测试中的标准工作负荷非常有帮助。
- 第 11 章介绍了缺少索引以及数据库优化顾问。
- 第 12 章介绍了存储子系统的性能和健壮性，并说明了如何通过使用 SQLIO 和 SQLIOSim 加大效果。
- 第 13 章介绍了 SQL Server Performance Dashboard 的使用，它能够帮助找出性能问题，而且是捕获性能量度的好的方式。

通过对这一部分的学习，应当清楚如何减少各种瓶颈的发生。

第IV部分 服务器性能路线图

这一部分讨论了关于如何达到更好性能的话题，还介绍了在产品生命周期的后期阶段中进行性能调校时遇到的一些挑战。

- 第 14 章介绍了从一开始就加入性能设计的最佳实践。
- 第 15 章介绍了成功部署的最佳实践。

使用本书的条件

为了能够实现本书中的例子，需要安装 SQL Server 2005。应当安装上最新的 Service Pack。在撰写本书时，最新的 Service Pack 版本是 2。

由于本书针对小规模 and 中等规模业务用户，因此可以使用 SQL Server Standard edition、Workgroup edition 或 Developer edition 来实现多数例子。SQL Server Workgroup edition 将不能够实现关于数据库优化顾问的例子。

不推荐使用 SQL Server Express 或 SQL Server Compact Edition。尽管很多例子也能够运行，但是我们试图说明的行为在这些 SQL Server 版本上有所不同。

如果使用的是 SQL Server Enterprise edition，则会在某些例子中得到的部分查询计划的细节方面发现一些区别，但是最终结果仍将保持一致。

您所使用的计算机应当能够满足所使用的 SQL Server 版本的最低硬件要求。在大多数情况下，本书中的例子是在运行 Windows Vista(RTM build 6.0.6000.2.0.0)的膝上型电脑上创建和测试的。

此外，还需要具有 T-SQL 的应用知识，而且知道如何执行基本的 SQL Server 和操作系统管理任务。

源代码

在读者学习本书中的示例时，可以手动输入所有的代码，也可以使用本书附带的源代码文件。本书使用的所有源代码都可以从本书合作站点 <http://www.wrox.com/>或 www.tupwk.com.cn/downpage 上下载。登录到站点 <http://www.wrox.com/>，使用 Search 工具或使用书名列表就可以找到本书。接着单击本书细目页面上的 Download Code 链接，就可以获得所有的源代码。

注意：

由于许多图书的标题都很类似，所以按 ISBN 搜索是最简单的，本书英文版的 ISBN 是 978-0-470-17639-9。

在下载了代码后，只需用自己喜欢的解压缩软件对它进行解压缩即可。另外，也可以进入 <http://www.wrox.com/dynamic/books/download.aspx> 上的 Wrox 代码下载主页，查看本书和其他 Wrox 图书的所有代码。

勘误表

尽管我们已经尽了各种努力来保证文章或代码中不出现错误，但是错误总是难免的，如果您在本书中找到了错误，例如拼写错误或代码错误，请告诉我们，我们将非常感激。通过勘误表，可以让其他读者避免受挫，当然，这还有助于提供更高质量的信息。

请给 wkservice@vip.163.com 发电子邮件，我们会检查您的反馈信息，如果是正确的，我们将在本书的后续版本中采用。

要在网站上找到本书英文版的勘误表，可以登录 <http://www.wrox.com>，通过 Search 工具或书名列表查找本书，然后在本书的细目页面上单击 Book Errata 链接。在这个页面上可以查看到 Wrox 编辑已提交和粘贴的所有勘误项。完整的图书列表还包括每本书的勘误表，网址是 www.wrox.com/misc-pages/booklist.shtml。

P2P.WROX.COM

要与作者和同行讨论，请加入 p2p.wrox.com 上的 P2P 论坛。这个论坛是一个基于 Web 的系统，便于您张贴与 Wrox 图书相关的消息和相关技术，与其他读者和技术用户交流心得。该论坛提供了订阅功能，当论坛上有新的消息时，它可以给您传送感兴趣的论题。Wrox 作者、编辑和其他业界专家和读者都会到这个论坛上来探讨问题。

在 <http://p2p.wrox.com> 上，有许多不同的论坛，它们不仅有助于阅读本书，还有助于开发自己的应用程序。要加入论坛，可以遵循下面的步骤：

- (1) 进入 p2p.wrox.com，单击 Register 链接。
- (2) 阅读使用协议，并单击 Agree 按钮。

- (3) 填写加入该论坛所需要的信息和自己希望提供的其他信息，单击 **Submit** 按钮。
- (4) 您会收到一封电子邮件，其中的信息描述了如何验证账户，完成加入过程。

注意：

不加入 P2P 也可以阅读论坛上的消息，但要张贴自己的消息，就必须加入该论坛。

加入论坛后，就可以张贴新消息，响应其他用户张贴的消息。可以随时在 Web 上阅读消息。如果要想让该网站给自己发送特定论坛中的消息，可以单击论坛列表中该论坛名旁边的 **Subscribe to this Forum** 图标。

关于使用 Wrox P2P 的更多信息，可阅读 P2P FAQ，了解论坛软件的工作情况以及 P2P 和 Wrox 图书的许多常见问题。要阅读 FAQ，可以在任意 P2P 页面上单击 FAQ 链接。

目 录

第 I 部分 在问题发生时寻找瓶颈

第 1 章 性能调校	3
1.1 艺术还是科学	3
1.2 性能调校的科学	4
1.2.1 问题陈述	4
1.2.2 处理计划	6
1.2.3 数据收集	6
1.2.4 数据分析	7
1.3 应用性能调校	7
1.3.1 实例 1: 应用的速度慢	7
1.3.2 实例 2: 存储过程 X 的 速度慢	8
1.3.3 工具	10
1.3.4 预防措施	13
1.4 小结	14
第 2 章 使用 System Monitor 监视 服务器资源	15
2.1 为什么可能会需要 System Monitor	15
2.2 何时应当使用 System Monitor	16
2.3 Performance Monitor 概述	16
2.4 开始使用 System Monitor	18
2.5 运行 System Monitor 的影响 是什么	22
2.5.1 对监视的影响进行管理	22
2.5.2 在恰当的时间, 以恰当的 持续时间进行捕获	23
2.6 System Monitor 将生成 多少数据	23
2.7 资源利用	24
2.8 识别瓶颈	25

2.8.1 定位内存瓶颈	25
2.8.2 发现磁盘瓶颈	26
2.8.3 识别 CPU 瓶颈	28
2.9 主动使用 System Monitor	29
2.10 在 64 位系统上运行 System Monitor	29
2.11 合并 System Monitor 日志和 SQL Profiler 跟踪	30
2.12 监视远程服务器	31
2.13 System Monitor 的最佳实践	31
2.13.1 获得一个基线	31
2.13.2 保留性能日志	32
2.13.3 模式和趋势	32
2.13.4 经受糟糕性能的服务器	32
2.13.5 调校性能	32
2.13.6 采取主动	32
2.14 System Monitor 计数器不见了 应当怎么办	33
2.15 内置日志管理工具	34
2.15.1 LogMan	34
2.15.2 Relog	35
2.16 分析日志数据	35
2.16.1 LogParser	35
2.16.2 Performance Analysis of Logs	36
2.17 小结	37
第 3 章 使用 System Monitor 监视 SQL Server 资源	39
3.1 开始	40
3.2 性能问题的类型	40
3.2.1 基于配置的性能问题	40
3.2.2 基于模式的性能问题	41

3.3	瓶颈的类型	42
3.4	内存瓶颈	43
3.4.1	内存瓶颈的类型	43
3.4.2	确认内存瓶颈	44
3.4.3	基于配置的内存瓶颈	45
3.4.4	基于模式的内存瓶颈	46
3.5	CPU 瓶颈	46
3.5.1	确认 CPU 瓶颈	46
3.5.2	基于配置的 CPU 瓶颈	47
3.5.3	基于模式的 CPU 瓶颈	48
3.6	磁盘瓶颈	49
3.6.1	确认磁盘瓶颈	49
3.6.2	基于配置的磁盘瓶颈	50
3.6.3	基于模式的磁盘瓶颈	53
3.7	监视数据库镜像性能	55
3.8	监视 Wait Statistics	55
3.9	典型性能问题	56
3.9.1	典型磁盘问题	56
3.9.2	内存	57
3.9.3	典型 CPU 问题	57
3.10	使用 SQL Server 来分析性能 日志	59
3.11	联合 Performance Monitor 日志 和 SQL Profiler 跟踪	63
3.12	小结	64
第 4 章	SQL Server 等待类型	65
4.1	SQL Server 等待	65
4.2	体系结构	66
4.3	常见的或值得注意的资源等待 类型	67
4.4	如何追踪等待	68
4.4.1	sys.dm_exec_requests: 只提供会话级信息	68
4.4.2	sys.dm_os_waiting_tasks: 所有等待任务	69
4.4.3	sys.dm_os_wait_stats: 根据等待类型聚集时间	69
4.4.4	DMVStats	70

4.4.5	Performance Dashboard	71
4.4.6	等待统计实战	71
4.5	加锁与阻塞	77
4.5.1	并发性	77
4.5.2	事务	78
4.5.3	隔离级别	79
4.5.4	锁和加锁	80
4.5.5	数据行版本控制	83
4.5.6	对死锁进行监视	84
4.5.7	对阻塞锁进行监视	87
4.6	小结	89

第 5 章 使用 SQL Profiler 寻找

问题查询	91
5.1 设置跟踪的准备工作	91
5.1.1 检测“问题”陈述 是否完备	91
5.1.2 寻找相关已知问题	92
5.1.3 以 SQL Trace 的术语 进行思考	92
5.1.4 SQL Trace 选项及思考	96
5.2 捕获阻塞事件	102
5.3 捕获 Showplan XML 数据	105
5.4 捕获死锁图	107
5.5 使用 SQL Profiler 识别长时间 运行的查询	112
5.5.1 模拟一个场景及示例 数据库	112
5.5.2 分析问题	112
5.5.3 设置	113
5.5.4 进行跟踪	115
5.5.5 分析和调校	116
5.5.6 案例总结	119
5.6 使用 Profiler 生成服务器端 跟踪代码跟踪高成本查询	119
5.6.1 使用 Profiler 生成服务器端 跟踪脚本	119
5.6.2 执行服务器端脚本	123

5.6.3 管理跟踪文件并分析跟踪数据.....	123	8.1.2 反规范化.....	189
5.6.4 服务器端跟踪代码演示.....	127	8.1.3 定义列.....	191
5.6.5 案例总结.....	133	8.1.4 存储过程.....	205
5.7 将 Profiler Trace 同 System Monitor 性能计数器数据关联起来.....	133	8.1.5 触发器.....	209
5.8 小结.....	135	8.2 数据性能.....	212
第 II 部分 通过调校去除瓶颈		8.2.1 页.....	212
第 6 章 选择和配置硬件.....	139	8.2.2 分区.....	215
6.1 服务器瓶颈.....	139	8.2.3 并发性与加锁(以及阻塞).....	218
6.1.1 内存.....	139	8.2.4 锁.....	220
6.1.2 I/O.....	140	8.2.5 索引.....	222
6.1.3 CPU.....	140	8.3 小结.....	227
6.2 配置服务器.....	140	第 9 章 调校 T-SQL.....	229
6.2.1 内存.....	142	9.1 开局: 优化器的策略.....	229
6.2.2 I/O.....	149	9.1.1 基于成本的优化.....	230
6.2.3 CPU.....	158	9.1.2 读取索引统计.....	230
6.3 小结.....	166	9.1.3 Include Actual Execution Plan 的误解.....	232
第 7 章 调校 SQL Server 配置.....	167	9.1.4 使用 sp_helpindex 来检查索引.....	232
7.1 调整服务器级设置之前的考虑.....	167	9.2 中局: 收集事实.....	232
7.2 检查当前 SQL Server 设置.....	168	9.2.1 查询计划.....	232
7.2.1 使用 SQL Server Management Studio 检查服务器设置.....	168	9.2.2 查询计划的实质.....	234
7.2.2 使用脚本检查服务器设置.....	169	9.2.3 从计划高速缓存中检索查询计划.....	250
7.2.3 检查 Advanced Settings 并修改它们的值.....	170	9.2.4 缓冲池.....	251
7.3 重要的服务器设置: CPU、内存和 I/O.....	171	9.3 终局: 对 T-SQL 进行性能调校.....	257
7.3.1 CPU.....	171	9.3.1 主要工具.....	257
7.3.2 内存.....	177	9.3.2 实例的必备索引.....	259
7.3.3 网络 I/O.....	178	9.4 使用新的方法来调校 T-SQL.....	259
7.4 小结.....	181	9.4.1 NOT IN 和 NOT EXISTS 重写已经成为过去.....	259
第 8 章 调校模式.....	183	9.4.2 通过将谓词在计划中推进到更深的位置进行重写.....	260
8.1 数据质量.....	183	9.4.3 为中间结果使用临时表.....	262
8.1.1 规范化.....	183	9.4.4 SELECT 语句中的用户定义函数.....	263
		9.4.5 改造 SELECT *.....	265

9.5	调校 T-SQL 谓词	269
9.5.1	去除特定隐式转换	269
9.5.2	使用可搜索参数	271
9.6	使用索引调校 T-SQL	274
9.6.1	最小化书签查找	274
9.6.2	对结果进行排序	276
9.6.3	处理编入索引的可空列	277
9.6.4	消除重复	278
9.7	调校 T-SQL 常见模式或 惯用语	280
9.7.1	单元素 SELECT 语句	280
9.7.2	聚集和 CTE	283
9.7.3	派生表和相关子查询	286
9.7.4	分页和排序	290
9.8	使用提示调校 T-SQL 语句	296
9.9	调校死锁和阻塞	299
9.9.1	阻塞	299
9.9.2	死锁	299
9.9.3	预防措施	300
9.10	对查询计划进化进行 模拟压力测试	303
9.11	小结	306

第 III 部分 使用工具来采取主动措 施并建立性能基线

第 10 章	使用 SQL Profiler 捕获、 测量、重放工作负荷	309
10.1	为了重放对工作负荷 进行捕获	309
10.1.1	为了重放对工作负荷 进行刻画	309
10.1.2	满足工作负荷重放的 要求	311
10.1.3	捕获工作负荷	313
10.1.4	为了特殊需要在跟踪表 中修改工作负荷	314
10.2	测量工作负荷性能	317
10.2.1	工作负荷的初步分析	317

10.2.2	用于工作负荷重放的 新的性能参考	321
10.3	重放工作负荷	323
10.3.1	为案例场景生成工作 负荷	325
10.3.2	场景 1: 验证性能 改进	326
10.3.3	场景 2: 在不同环境中 重放工作负荷并测量总 体响应时间	329
10.3.4	场景 3: 在目标服务器上 重放多个工作负荷	332
10.4	小结	337

第 11 章	调校索引	339
11.1	示例数据库	339
11.2	第 1 部分: 索引回顾	341
11.2.1	B 树	341
11.2.2	填充因子	342
11.2.3	聚簇索引	342
11.2.4	堆	344
11.2.5	非聚簇索引	345
11.2.6	统计	346
11.2.7	加索引的视图	350
11.3	第 2 部分: 用 DTA 进行 调校	350
11.3.1	使用 DTA 调校个别 的查询	351
11.3.2	用于更新的索引	360
11.3.3	增加更新索引后重新 评定插入	372
11.3.4	索引过多	375
11.3.5	调校工作负荷	378
11.4	第 3 部分: 索引维护	380
11.4.1	监视索引碎片	380
11.4.2	删除碎片	381
11.5	第 4 部分: 分区的表和 索引	383

11.5.1	使用分区的表和索引 的理由	383	13.2.1	必要条件	457
11.5.2	分区的先决条件	384	13.2.2	安装 Performance Dashboard Reports	457
11.5.3	创建分区的表	384	13.2.3	运行 Performance Dashboard Reports	460
11.6	小结	394	13.2.4	架构	463
第 12 章	存储的速度及健壮性	395	13.2.5	常见使用场景	465
12.1	性能测试、压力测试和实际 性能	395	13.2.6	局限性	469
12.1.1	性能测试	395	13.3	相关链接	471
12.1.2	压力测试	396	13.4	小结	471
12.1.3	实际性能	396			
12.1.4	汇总	397	第 IV 分	服务器性能的路线图	
12.2	存储性能	397	第 14 章	从一开始就为性能进行设计 的最佳实践	475
12.2.1	存储性能测量工具	397	14.1	理解性能需求	475
12.2.2	希望测量什么	398	14.1.1	数据库将支持多少 用户	476
12.2.3	测量 I/O 性能	400	14.1.2	用户/角色访问要求 的本质	478
12.2.4	使用 SQLIO	408	14.1.3	业务事务要求的 吞吐量	479
12.3	存储的可靠性	434	14.1.4	软件应用程序的核心 架构	481
12.3.1	SQLIOSim	434	14.2	为未知的性能评估模式	482
12.3.2	使用 SQLIOSim	434	14.2.1	简化模式	483
12.3.3	SQLIOSim 测试持续时间 的最佳实践	437	14.2.2	单一用途表	483
12.3.4	运行其他测试	438	14.2.3	单一用途列	484
12.3.5	从命令行运行 SQLIOSim	446	14.2.4	消除不明确的空字段	486
12.3.6	解释来自 SQLIOSim 的 结果	446	14.2.5	基于 ID 的分区	487
12.4	小结	449	14.2.6	为了性能进行 反规范化	487
第 13 章	SQL Server 2005 Performance Dashboard Reports	451	14.2.7	视图中函数的性能 影响	489
13.1	可支持性	451	14.3	为性能评估索引	490
13.1.1	目录视图	452	14.3.1	不加思考/默认索引	490
13.1.2	动态管理视图及函数	452	14.3.2	简单访问模式索引	491
13.1.3	默认跟踪	453	14.3.3	复杂访问模式索引	491
13.1.4	困难	456	14.4	为评估建立基准	491
13.2	Performance Dashboard Reports	456	14.4.1	创建基准模式	491

14.4.2	从 DMV 中提取 Object_ID 级信息	492	15.6.1	分区函数	506
14.4.3	捕获表吞吐量	492	15.6.2	分区方案	506
14.4.4	监视索引使用	492	15.6.3	对表或索引进行分区	507
14.4.5	TempDB 使用	493	15.7	使用 SQL Server Profiler 进行调校	507
14.4.6	捕获阻塞信息	494	15.8	使用 Database Engine Tuning Advisor 进行调校	507
14.4.7	监视高 CPU 使用	495	15.9	错误预估实际执行环境 的后果	507
14.4.8	监视高 I/O 产生者	496	15.10	实际执行中的模式问题	508
14.5	沟通性能问题	497	15.11	避免通过修改代码来 解决问题	509
14.6	小结	498	15.11.1	查询	509
第 15 章	成功的部署策略	499	15.11.2	查询计划	510
15.1	预估	500	15.11.3	Plan Guide	511
15.2	附加特性	501	15.11.4	创建 Plan Guide 之后的 查询计划	511
15.3	高可用性以及灾难恢复	501	15.12	小结	513
15.3.1	备份和恢复	502			
15.3.2	集群	502			
15.3.3	数据库镜像	503			
15.3.4	日志传送	503			
15.3.5	复制	503			
15.4	负荷测试	504			
15.5	管理变动	504			
15.6	处理大的表	505			

第 I 部分

在问题发生时寻找瓶颈

第 1 章 性能调校

第 2 章 使用 System Monitor 监视服务器资源

第 3 章 使用 System Monitor 监视 SQL Server 资源

第 4 章 SQL Server 等待类型

第 5 章 使用 SQL Profiler 寻找问题查询

第 1 章

性能调校

SQL Server 2005 是即装即用的，它的默认安装可以满足大部分用户的需要。对于大多数 SQL Server 用户，从来不需要考虑性能问题。他们只需要把 SQL Server 安装起来，加载数据，系统就能很好地工作。但是，在过去的几年里，数据库规模以及小规模及中等规模业务可使用的硬件性能都有了爆炸式的增长，这已经使得更多的用户能够构建日益庞大的系统。当前的很多系统在规模上甚至超过了过去要求使用大型机级别硬件的企业级系统，这种规模及性能上的增长令更多的用户不得不尝试突破 SQL Server 默认设置的性能限制。这意味着更多的 SQL Server 用户必须开始学习 SQL Server 中他们本不需要了解的方面——性能调校。

好消息是正如即装即用的体验一样，SQL Server 开发小组投入了大量的工作，使得 SQL Server 成为一个易于进行性能调校的数据库系统。SQL Server 的竞争者们往往需要高度受训且收费昂贵的咨询人员来诊断并调校性能问题，和这些竞争者不同，SQL Server 有着可用的工具和信息，使得任何人都可以对他们的数据库进行调校。

1.1 艺术还是科学

一直以来，性能调校都被一种神秘的光环所围绕。在未掌握这种技能的人看来，少数掌握这种技能的性能调校师是令人敬畏和好奇的。对于性能调校师圈子以外的人而言，往往会感觉需要很多年才能获得成为性能调校师所需具备的技能。

真实情况是只要有了系统的方法、适当的工具以及恰当的知识，任何对 T-SQL 有基本了解的人都能逐步采取措施来提高数据库系统的性能。当您开始掌握这些技能，就会发现性能调校不再像一种深奥的艺术，而是一门聚焦性很强的科学。

本书的目的正是为了给您提供在开始以系统的方法进行性能调校时所需的知识，并告诉您使用何种工具以及何时使用这些工具，为您提供成为一个成功的 SQL Server 性能调校师所需的知识。