

数 · 据 · 库 · 专 · 家 · 系 · 列

胡百敬 著

Microsoft®

SQL Server

性 能 调 校

- ◆ 微软专业顾问、台湾恒逸资讯资深讲师经典力作
- ◆ 繁体版原作大受好评
- ◆ 性能调校圣经
- ◆ 深入剖析 SQL Server 的运作原理



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

Microsoft®
SQL Server 性能调校

胡百敬 著

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

SQL Server 是当前应用最广的大型数据库系统之一,大型数据库系统运行一段时间后就会出现运行缓慢、性能下降、故障增多等问题,为了使系统维持正常运行,必须对系统不断进行“调校”。

本书从 SQL 系统的运行原理入手,探究其常遇运行问题的形成原因与解决办法。本书作者长期担任大型数据库设计开发和管理的顾问并从事 SQL Server 系统的教学工作,本书是他丰富的实践与教学经验的结晶。全书内容包括:性能调校概观、SQL Server 架构简介、性能调校相关工具程序、数据库设计、T-SQL 语法、索引与查询性能、事务与锁管理、前端应用程序设计。

无论是正在从事大型数据库系统的设计、开发、或教学工作的人员或者是在学习大型数据系统知识的学生,掌握“系统调校”知识都是必不可少的。

图书在版编目(CIP)数据

Microsoft SQL Server 性能调校 / 胡百敬著. —北京: 电子工业出版社, 2005.1
ISBN 7-121-00438-0

I.M… II.胡… III.关系数据库—数据库管理系统, SQL Server IV.TP311.138

中国版本图书馆 CIP 数据核字(2004)第 103691 号

责任编辑: 周 筠 陈元玉

技术协作: 刘铁锋

责任校对: 张兴田

印 刷: 北京智力达印刷有限公司

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

经 销: 各地新华书店

开 本: 787×980 1/16 印张: 25.75 字数: 560 千字

印 次: 2005 年 1 月第 1 次印刷

印 数: 5 000 册 定价: 48.00 元(含光盘 1 张)

凡购买电子工业出版社的图书,如有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系。联系电话:(010) 68279077。质量投诉请发邮件至 zlts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

推荐序

SQL Server 2000 上市至今，已经成为企业通用的数据库软件，大概所有企业都会使用到 SQL Server，因此对于 SQL Server 数据库系统的管理与数据库程序的编写(T-SQL)，相信已经有很多人熟悉。然而在熟悉 SQL Server 的管理与程序设计之后，数据库将会有有一个最终极的问题，那就是性能调校，如果你不熟悉数据库管理，对数据库程序设计也没有概念，那性能调校对于你而言无须考虑；但若是该做的你都做了，接下来想了解该怎么办时，你就得要好好研究性能调校了。

数据库本身也有生命周期，它并不是一构建好就可以放在旁边永远不用再去理它的，若是你对性能的要求更高，那你很可能得在数据库还没开始构建之前就必须考虑将来可能会出现性能“瓶颈”的问题，性能这个问题是全面性的问题，其所涵盖的范围非常广，不单单是懂得数据库管理与设计就可以解决的。

若是性能问题一直在困扰着你，那么本书便可以为你解惑。本书作者胡百敬先生拥有多年的实际大型数据库顾问经验，且本身经常担任 SQL Server 的技术讲师，技术功底深厚，凭借他多年的经验与研究心得，撰写出这本针对性能调校议题的书籍，相信能对顾虑 SQL Server 性能问题的读者有所贡献。

张智凯

Richard Chang

台湾恒逸信息教育训练处
系统开发部经理

自序

Microsoft SQL Server 已经是世界上应用最普遍的大型数据库之一，在许多系统中都有 SQL Server 的踪迹。而每个使用数据库的人，都可能会面临到数据累积、使用人数增加、当初系统设计或程序撰写不良等各种原因，导致性能变差。让用户无法忍受系统响应的时间，待处理的工作长期停滞在系统的队列中，甚至是程序无法执行，系统崩溃，数据大乱。然而信息系统已经是一般企业不可或缺的竞争力之一，而数据库系统又往往是信息系统的核心，跑不动的数据库绝对是莫大的伤害。

数据库管理员(DBA)或系统开发人员在发生性能问题的初期，大都没有方向，不知如何解决问题。市面上有一堆书籍教人如何利用 SQL Server 建立数据库，或是以 Visual Basic 等程序语言存取数据库。其实，数据库一下子就建好了，经过个把月程序也写完了，但今后数年都需要对数据库、对系统进行性能调校。

性能调校的能力是需要培养的，不是做了多年的 DBA 就一定可以解决各式各样层出不穷的问题，因为现在的系统都已经太过复杂，任何一个小环节发生的问题，在交互影响下，显示出来的现象往往会误导我们的判断，让调校人员耗费了非常多的时间与精力，但往往是一场虚功。

性能调校不是一件简单的事，一般来说需要有广泛的经验与知识，不单单是数据库的经验，还要对商业领域知识、系统架构设计，应用程序撰写，操作系统，网络环境架设，各种侦测与监控工具程序的使用等等，都需要有基本的了解，才能在复杂的系统中，找到症结所在。知道问题后，并不一定有解，如何简便而有效地解决问题又是另一项挑战。

信息技术不断更迭创新，而应用层面广泛多变，笔者碰到来自各方的朋友讨论系统的“瓶颈”所在，但往往因为需求不明，应用领域不熟，很难有具体的结果。西谚“One man's meat is another man's poison(你喜好的肉糜对我却是毒药)”恰能描绘性能调校的技巧难以普遍适用的困境，我们所讨论的功能、技法往往是对某些系统应用好，但在另一些情境下则适得其反。我听过很多的反映，例如：

- 原本以为通过 ADO.NET 的 Dataset 可以解决 SQL Server 的瓶颈，没想到先拖垮了 IIS。
- 本想将数据缓存到 COM+ 以减少 SQL Server 的存取，没想到 COM+ 先挂了。
- 本以为减少数据库联机的程序撰写方式可以节省服务器资源，没想到反而发挥不出多线程、多 CPU 的运算能力。
- ...

Microsoft SQL Server 性能调校

诸此种种，不可胜数。系统设计的好坏，存乎一心，架构好，系统可大可久，架构不好，处处捉襟见肘。但何谓好何谓不好，没有定论。只有你小心翼翼测试采用的技术在自己的商业领域中，应用得是否恰当。

本书尝试定义、找寻与解决问题。深入探究 **SQL Server** 的运作原理，如索引、游标、数据存取接口、事务与锁等等。我们将查看与仿真 **SQL Server** 运行时与性能相关的议题，通过工具来探索问题的来龙去脉。例如大量的用户同时存取时，系统上有太多的锁/阻塞/死锁(lock/block/deadlock)，其形成原因、如何跟踪、如何解决，诸如此种大部分数据库管理员可能天天面临的问题。而系统设计时应注意的事项，包括程序该如何撰写，各数据库对象使用方式的优劣也是本书的重点之一，务求有一个好的数据库设计，也有一个好的应用程序来存取它。笔者希望能以多年使用 **SQL Server** 的经验，参考并整理散在诸多微软标准教材、网站、书籍与文献中的资料，期待能为你稍解性能调校这个恼人的问题。

本书所示范各种性能调校的技巧都是执行在 **MS Windows 2000**、**XP** 以及 **Windows 2003** 操作系统上，执行的 **SQL Server** 版本为 **2000**。你所使用的操作系统或是 **SQL Server** 的版本或许不同，但如何建立一个有效率的系统，当应用程序性能不足时，要如何调校？其中的概念、使用的工具与大部分的技巧都是相通的。因此就算你所使用之环境与前述软件的版本或有不同，本书仍应该具有参考价值。

这是一本 **SQL Server** 进阶的书籍，是提供给已经在使用 **SQL Server** 的工程师一些性能调校或错误处理的建议与提示，因此一些基本的操作，或是应有的数据库基本观念笔者都未解释，仅列出请你参照的资源。

笔者常常与来恒逸听课的朋友谈到：“技术可以教，而艺术不能教”。性能调校比较偏向于艺术，因为有太多技巧是两面刃，本可能有益于系统，但条件稍有不同时，却反对系统有害。由于小而零散的规则极多，笔者极企盼能从更深的层面找到共通运行的原理，让你可以收提纲挈领之效，无奈才疏学浅，看到了一堆的点，却没有贯穿的线。这让笔者一年来一直犹豫要如何完成这本书，眼见时间匆匆流逝，新技术滚滚而来，既要深入已有的技术，又要融合新的趋势。唉，凭一己之力在信息长河只能取一瓢饮，难有宏观擘画的能力，此时完成此书让它面市，是希望对当下的技术完成一个快照，提供大家现行系统性能分析的基点。

感谢联经信息的蓝春华、潘如莹、黄振祖、周建龙、薛凤玉、陈志玉等同仁的帮忙与协助，长期以来，我们一次又一次地完成了性能调校的任务，愿以此书留下我们努力的脚印。同时也谢谢亚洲志远的云志文、黄志尧两位先生愿意提供为联合报系做公文签核流程系统的压力测试文件，笔者将其略为简化后收录在附录 **B** 当做参考模板。

沁凉的深夜，从书桌前抬起头，回首熟睡的妻儿，总有一股淡淡的惆怅与歉意。每一本书的写成是自我对技术里程的脚注，也是对家人怠慢与轻忽的集成。

感谢慧无私的宽容与支持，谨将此书献与妻。

导 读

本书共分八个章节以及四个附录，彼此之间有相当的独立性，你可以依需求单独挑选部分章节研读。书后所附的光盘中对应各章的内容，提供相关的范例源程序代码以及工具程序，辅助你深入了解文中所述的各种概念。

以下就各章节的内容做一个概略的描述：

第 1 章 性能调校概观

性能调校不是一件简单的事，一般来说需要有广泛的经验与知识，不单单是数据库的经验，还要对系统架构设计，应用程序撰写，操作系统，网络环境架设，各种侦测与监控工具程序的使用等等，都应当有基本的了解，才能在复杂的系统中，找到症结所在。

在本章中主要定义什么是性能调校、为何要做性能调校以及如何完成性能调校，让你每每在面对不同的情境，庞杂的工作中有所脉络可循，依照一定的步骤与进程来拟定性能调校计划，完成性能调校的工作。

第 2 章 SQL Server 架构简介

要对 SQL Server 开始做性能调校前，应当先了解 SQL Server 本身的架构，以及 Windows NT 系列操作系统的运行原理，如此才容易界定问题发生的原因，并找出因应的对策。

在本章中将介绍 SQL Server 的组成与执行方式，系统软硬件架构对 SQL Server 的影响，以及 SQL Server 使用 CPU、内存与硬盘的方式，有哪些设定可供调整，以及哪些性能计数器(performance counter)可资参考。

第 3 章 性能调校相关工具程序

当你要对 SQL Server 做性能调校时，需要依不同的状况善用各种工具程序，做广泛的分析和深入的研究。

在本章中将介绍进行性能调校时可采用的相关工具程序，如 Performance Monitor、SQL Enterprise Manager、SQL Query Analyzer、T-SQL Script、Profiler、DBCC、跟踪标记、网络包捕获程序以及应用程序压力测试工具等等。经由这些工具程序可以让你从广

泛地观察问题的全貌开始，到深入剖析关键核心所在。

第4章 数据库设计

数据库的设计千头万绪，要从逻辑的数据使用对应到物理的数据存放，其中要考虑功能、效率、安全、稳定性、扩充性、容易存取与维护等等诸多因素，导致设计数据库很难一次完成。

在本章中将介绍有关数据库容量、范式与否、联机事务处理(OLTP)与联机事务分析(OLAP)在设计上的差异、备份(Backup)与还原(Restore)与数据库文件配置等等相关议题。

第5章 T-SQL 语法

由于 SQL Server 只接受 T-SQL 语言，它的目的就是要完成你所给的 T-SQL 之语意，所以善用 T-SQL 语法是善用 SQL Server 的基础。但因为笔者撰写本书的目的并不是讨论 T-SQL 这个语言本身，所以在本章仅讨论与性能较有关系的部分。

在本章中将介绍查询参数(search argument SARG)、SQL Server 内建的各种公共变量、联机设定、需要注意的语法，以及提供一些工具性的 SQL Script，可以辅助你执行与性能相关的工作。

第6章 索引与查询性能

一般索引建立的适当与否，是性能好坏的成功关键。并由于更改索引结构不会影响数据字段定义，也就是前端程序可以照常存取，因此当上线后的程序性能表现不好时，尝试最佳化索引往往是第一个考量。

在本章中将介绍索引的运行原理、索引的建立时机、增加索引的 T-SQL 语法与相关设定、有效使用索引的方式、SQL Server 如何以成本计算为基础(cost base)判读合用的索引、统计的功用与重要性、SQL Server 所提供的联结(Join)机制与查询效率，以及 Indexed View 的使用等等。

第7章 事务与锁管理

SQL Server 要依靠事务管理来维护商业逻辑的正确性，通过锁(Lock)的机制来让更多人存取的资源不至于紊乱。让大家都在抢资源时，可以有先来后到的顺序。但也因为要排队等资源，这往往让众人卡在某笔未被释放记录，整个系统看起来好像当掉。虽然系统有非常强悍的软硬件，但若应用程序使用事务与锁的方式不佳，一切都是枉然。

在本章中将介绍事务与锁的运作原理，如锁的种类、事务隔离层级等等。同时解释批处理与事务的关系、何谓巢状事务、死锁的种类与预防、系统当前锁的观察与分析等议题。

第 8 章 前端应用程序设计

在应用程序开发的过程中就需要注意性能问题，时时仿真真实的数据与取用方法，务求在规划时，就能兼顾数据的正确性与存取效率。越早知道效率瓶颈，越有可能完美地解决，所花的成本也越低。

在本章中将利用 **Visual Basic** 与 **Visual Basic.NET** 等语言，介绍应用程序存取 **SQL Server** 的各种方式，以及何谓 **Connection Pooling**、探讨游标的使用与应用程序的错误处理等等。

附录 A 与 B 分别提供笔者在任职顾问的公司所做的性能调校与压力测试计划之模板，它们的内容是针对特定的系统而定，所以并非完善的计划。笔者放入附录，目的是提供一个参考，你可以依此建立自己系统所需的计划。附录 C 列出笔者在撰写本书时所参考的资料来源，不管是从网站、书籍或是微软的教材，都可以让你进一步了解更多与系统性能相关的知识。最后在附录 D 笔者稍微描述了 64 位系统在当下这个时间点的定位，希望你评估系统时有所帮助。

目 录

推荐序	i
自 序	iii
导 读	v

第 1 章 性能调校概观

1.1 什么是性能调校	5
1.2 建立性能的基线及相关文件	5
1.3 性能调校的步骤流程 - DETECT	8
1.3.1 各阶段重点说明	9
1.3.2 练习 DETECT 方法	13
1.4 定义瓶颈	16
1.5 结语	17

第 2 章 SQL Server 架构简介

2.1 SQL Server 运行架构	21
2.1.1 SQL Server 的存取架构	24
2.1.2 SQL Server 的核心引擎	27
2.1.3 动态自我管理	31
2.2 各项硬件使用剖析	33
2.2.1 内存管理	33
2.2.2 中央处理器	48
2.2.3 硬盘子系统	54
2.3 仿真系统运作	61
2.4 本章参考资源	63

第 3 章 性能调校相关工具程序

3.1 综观的工具	69
3.1.1 SQLDiag 概观	69
3.1.2 观察影响效率的属性	74
3.1.3 性能监视器	78
3.2 进一步的分析工具	84

3.2.1 SQL Server Enterprise Manager 及 T-SQL Script	84
3.2.2 SQL Profiler 概观	87
3.2.3 相关的系统对象	99
3.3 针对特定对象的工具	100
3.3.1 Query Analyzer	100
3.3.2 网络监视器	103
3.3.3 DBCC	106
3.3.4 跟踪标记	109
3.4 压力测试工具程序	110
3.4.1 Microsoft Application Center Test	111
3.4.2 自行撰写压力测试程序	114
3.5 本章参考资料	117

第4章 数据库设计

4.1 数据库设计	121
4.1.1 硬盘子系统之设计	126
4.1.2 分割 & 并行运作	130
4.2 备份与还原	134
4.2.1 Logical Log Marks	134
4.2.2 恢复模型	137
4.3 大量数据加载	139
4.4 结语	140

第5章 T-SQL 语法

5.1 有效的查询参数	143
5.1.1 不要对数据字段做运算	144
5.1.2 不要用负向查询	146
5.1.3 不要对数据字段使用函数	146
5.1.4 使用 OR 运算符要小心	147
5.2 使用 T-SQL 的注意事项	148
5.3 自行撰写管理用的辅助存储过程	150
5.3.1 当做工具的 T-SQL 程序	150
5.3.2 将存储过程建立在 tempdb 系统数据库	156
5.4 SQL Server 提供的公共变量	157
5.5 结语	165

第6章 索引与查询性能

6.1 索引概观	169
6.1.1 建立索引与相关的属性配置	170
6.1.2 聚集索引(Clustered Index)与非聚集索引	175
6.1.3 排序	179
6.1.4 并行建立索引	184
6.1.5 sysindexes 系统数据表	184
6.1.6 是否值得建索引	188
6.2 建立最优化执行计划的各阶段	191
6.3 统计	193
6.3.1 更新统计	200
6.4 联结与查询效率	204
6.4.1 联结	204
6.4.2 Join 语法	205
6.4.3 巢状循环联结	208
6.4.4 合并联结	209
6.4.5 哈希联结	210
6.4.6 联结与子查询	211
6.5 覆盖索引	213
6.6 在视图与计算字段上建立索引	217
6.6.1 如何有效地建立 Indexed View	219
6.6.2 Indexed View 的适用范围	224
6.7 查询提示	225
6.8 单一查询使用多个索引	230
6.9 数据不连续(Fragmentation)	231
6.10 结语	234

第7章 事务与锁管理

7.1 锁	237
7.1.1 锁的种类及范围	237
7.1.2 锁的兼容性	240
7.1.3 可锁定的资源	241
7.1.4 锁与事务隔离等级	243
7.1.5 动态的锁定管理	248

7.1.6 锁定超时	249
7.2 事务	252
7.2.1 批处理与事务	252
7.2.2 巢状事务	255
7.2.3 储存点	259
7.2.4 锁定提示	261
7.3 死锁	264
7.3.1 分布式死锁(Distributed Deadlock)	266
7.3.2 SQL Server 无法侦测到的死锁实例	268
7.4 观察与分析系统的锁定状况	273
7.4.1 观察 SQL Server 当前执行的状况	274
7.4.2 观察与分析系统的锁定状况	281
7.4.3 死锁状况分析	283
7.5 一般产生阻塞的原因	289
7.5.1 费时的查询或事务	289
7.5.2 不正确的事务或事务隔离级别配置	290
7.5.3 事务未正确处理	290
7.5.4 未侦测到的分布式死锁	291
7.5.5 Lock Granularity 太高或太低	291
7.5.6 Compile Blocking	291
7.5.7 防止锁住他人的技巧	292
7.5.8 防止与处理死锁的技巧	293
 第 8 章 前端应用程序设计	
8.1 处理 SQL 语法时用户端与服务器的互动情形	297
8.1.1 用户端存取 SQL Server 的模式	299
8.1.2 准备再执行的模式	303
8.1.3 通过 Master.dbo.Syscacheobjects 系统数据表观察在缓存中的对象	307
8.1.4 测试各种执行 SQL 语法方式的性能	312
8.1.5 Connection Pooling	313
8.2 游标	322
8.2.1 游标综述	322
8.2.2 默认结果集	324
8.2.3 服务器端游标	326
8.2.4 使用游标的 T-SQL 语法	331

8.2.5 与游标相关的系统存储过程	340
8.2.6 通过前端程序存取四种类型的游标	342
8.2.7 异步使用游标	346
8.2.8 使用游标时, 应注意的事项	348
8.3 应用程序错误处理	350
附录 A 性能调校计划	353
附录 B 压力测试计划	359
附录 C 参考资源	377
附录 D Wintel 的 64 位架构	383

第 1 章

性能调校概观

多年使用数据库的经验告诉我，数据库的性能从系统分析，应用程序设计开始，就需要时时小心，因为若基础不好，日后要修正数据库会因为牵涉到相关的前端应用程序，而大幅增加调校的复杂度。因此，不能等到系统做出来后才开始试试看，是否可以经得起多人使用且大数据量的压力。

性能调校不是一件简单的事，一般来说需要有广泛的经验与知识，不单单是数据库的经验，还要对系统架构设计、应用程序撰写、操作系统、网络环境架设、各种侦测与监控工具程序的使用等等，都应当有基本的了解，才能在复杂的系统中，找到症结所在。因为性能问题往往不会是单纯地显示出来的，让你一眼就知道答案。最简单的例子，当你看到 CPU 很忙，长时间处于百分之百的使用率，可能直觉就是 CPU 性能不足。但其实可能是硬盘性能不足，或是内存容量不够，让 CPU 一直忙于 I/O。这个例子只是简单地告诉你，**性能问题的症结通常不是表象所显示出来的。从不同的方面去看，往往会有不同的解释。**

微软的性能调校制式文件上标示着各项成功因素所占的百分率（见图 1.1）：先前的经验占 19%、解决问题的能力占 22%、是否有完备的顾问服务占 16%、产品的熟悉程度占 26%、计算机的相关知识占 13%、运气占 4%。天啊，你是全世界最强的信息人员，若上苍不眷顾，还是有 4% 的机会不能成功。

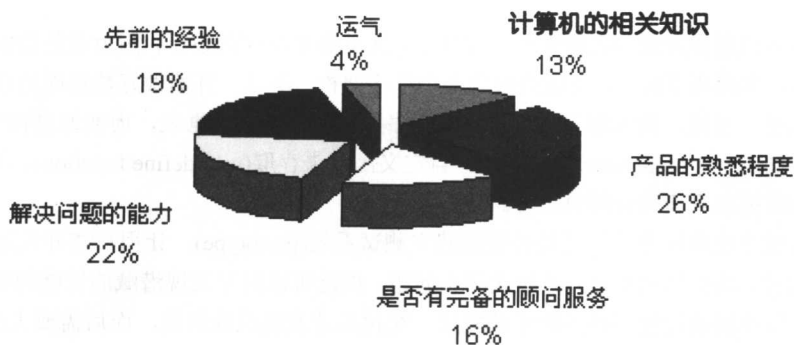


图 1.1 在解决问题时，各项成功因素所占的比例

而性能调校的能力是需要培养的，不是一个做了多年的 DBA 就一定可以解决各式各样层出不穷的问题的，因为现在的系统都已经太过复杂，任何一个小环节发生的问题，在交互影响下，显示出来的现象往往会误导我们的判断，让调校人员耗费了非常多的时间与精力，但往往是一份虚功。

因此要做好性能调校，你需要期许自己的知识不但要专精，且要对相关的知识保持高度的兴趣，可以广泛涉猎各个方面的技能，同时能够与人合作，一起找出问题，一个人单打独斗的时代早已经过去了。笔者常常参与不同的开发团队，一起做性能调校的工作，往往是某个人突