HZ BOOKS
华章科技

数据库
技术丛书

SQL Server 性能优化与管理的艺术

黄钊吉 著

The Art of SQL Server Performance Tuning and Management

- 深入剖析SQL Server的优化与管理机制及核心技术，从“方法论”的角度指导读者定位和解决问题，以独特视角展示SQL Server管理之道。
- 收录了与SQL Server性能优化相关的各类问题和工作中的“雷区”并给出了解决方法，包含大量技巧和最佳实践。

机械工业出版社
China Machine Press

数据库
技术丛书

SQL Server 性能优化与管理的艺术

The Art of SQL Server Performance Tuning and Management

黄钊吉 著

机械工业出版社
China Machine Press

图书在版编目（CIP）数据

SQL Server 性能优化与管理的艺术 / 黄钊吉著 . —北京：机械工业出版社，2014.8

ISBN 978-7-111-47324-4

I. S… II. 黄… III. 关系数据库系统 IV. TP311.138

中国版本图书馆 CIP 数据核字（2014）第 153692 号

全书共 15 章，分为三部分，第一部分（第 1 ～ 2 章）为概述部分，阐述 SQL Server 方面的“性能”及相关概念。并给出常规的性能及性能相关的问题侦测的“方法论”，读者可以通过这两章的介绍，对 SQL Server 性能问题有一个高层次的认识。第二部分（第 3 ～ 10 章）为知识准备部分，这部分介绍了 SQL Server 性能相关的基础知识。只有了解了性能及影响性能的相关部分，才能准确地、高效地进行优化。第三部分（第 11~15 章）为工具使用及优化演示，在多服务器、大数据的环境下，不应该再使用原始的故障侦测方法，借用各种工具能更全面、更高效地找到问题并且解决问题。

通过这三部分的介绍，可以使读者有一个清晰的性能优化及管理方面的认识，并且通过大量演示，让读者能够较快地进入实战阶段。本书的重点主要集中在第二部分，既介绍了性能相关的部分，以便后续使用，也给出了一个后续深入学习的“清单”，读者不应止步于本书，可以根据书中的知识点，进行更深入的学习。

SQL Server 性能优化与管理的艺术

出版发行：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　　责任校对：董纪丽

印　　刷：蓟城市京瑞印刷有限公司　　版　　次：2014 年 9 月第 1 版第 1 次印刷

开　　本：186mm × 240mm　1/16　　印　　张：31

书　　号：ISBN 978-7-111-47324-4　　定　　价：89.00 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　　投稿热线：（010）88379604

购书热线：（010）68326294　88379649　68995259　　读者信箱：hzjsj@hzbook.com

Preface 前 言

为什么要写这本书

大学时我就已经开始接触 SQL Server 了，而真正将它用于工作也有 6 个年头了。不管别人对它的评价如何，我依旧深爱着它。工作之后，接触到的 SQL Server 性能问题越来越多，也让我越来越着迷。后来我成为了专职的 DBA，对性能问题的探索也从兴趣变成了工作职责。

这几年来一直在学习 SQL Server，看了很多有关书籍，可真遇到性能问题的时候，还是一下子就蒙了，望着书架上那几十本书，不知该从哪里找类似的问题。其实很多人都有类似的经历，在看到 CPU 利用率很高、内存占了很多、整个 SQL Server 很慢时，常常不知道该怎么着手找到根源并处理。这也就成了我写这本书的一个初衷，我希望给出一个方法论，并给出一些后续学习的“大纲”，使读者运用书中的知识，去处理工作中遇到的绝大部分性能问题。

关系数据库发展到今天，各个关系数据库管理系统之间的差异已经很小，只是实现方式可能有所不同。基于这种情况，对于 DBA 来说，更重要的不是如何操作，而是掌握一个方法论和思路，有了方法论和思路，不管是对 Oracle 还是 MySQL，或者其他数据库管理系统，都可以实施优化，解决性能问题。找到问题的根源，处理问题，然后再检查、再处理，不管是数据库性能优化还是其他行业、领域的问题侦测，这个流程都是可以直接套上去或者稍作修改即可使用的。这也是本书的一个特色，不仅告诉读者 SQL Server 相关知识，还提供了基本方法和思路。

SQL Server 是世界级企业数据库领域的拳头产品，想要透彻了解它的每一个细节几乎不可能，但是我们即使不知道源代码，也可以很好地利用 SQL Server 来实现业务需求。我基于个人的工作经历进行写作，并且参考了大量国内外资料，以实用为目的来介绍知识。书中涉

及的知识点较多，其中一些展开足以独立成书。所以，这里更多的是给出一个“提纲”，让读者知道“有这个知识点”，如果读者有兴趣深入了解，可以去学习相关的专门资料。

读者对象

作为一本非入门书籍，本书假定读者已经有了一定的 SQL Server 基础和经验。本书主要面向的读者群体如下。

- 初中级 SQL Server DBA
- 中高级 SQL Server 相关程序员
- 使用 SQL Server 的项目负责人
- SQL Server 爱好者
- 对关系数据库性能优化有兴趣的读者
- 有志成为 DBA、数据库设计、数据库架构的 IT 从业人员

如何阅读本书

本书分为以下 3 部分。

第一部分是概述部分，介绍一些与性能相关的概念。

第二部分是与 SQL Server 性能相关的一些理论知识，为后续的使用打下基础。

第三部分主要是工具的使用，在了解了相关知识之后，就要学会如何去用。作为一直以界面友好著称的微软产品，SQL Server 同样提供了很多快速定位问题甚至直接指出问题的工具，供开发人员使用。

如果你是一名有少许经验的初学者，建议按顺序学习本书。如果你已经有一定的基础，可以从第二部分开始学习。另外，本书也可以作为开发人员外理工作中常见问题的一个“手册”。

勘误和支持

由于作者的水平有限，编写的时间也很紧迫，书中难免会出现一些错误或者不准确的地方，恳请读者批评指正。如读者发现有错误或者有疑问，欢迎发邮件到本人邮箱：huangzhaoji@hotmail.com，或者在本人的微博上留言，微博地址：http://weibo.com/u/3187132617。书中的一些重要脚本或者篇幅比较长的脚本将发布在机械工业出版社的华章公司网站（www.hzbook.com）上供读者下载，很期待能够听到读者们的真挚反馈。

致谢

我首先要感谢我的家人，特别是我的未婚妻李琳玲。写书占用了我几乎所有的休息时间，陪家人的时间基本上变成了“0”。但是他们一直理解我，并在工作和生活起居上为我打理一切。他们的辛勤付出为这本书的撰写创造了良好的环境。

然后我要感谢机械工业出版社给了本书出版的机会，特别感谢机械工业出版社华章公司的编辑杨绣国老师，感谢她在成书过程中对我的全力支持。也因为出书，逼自己学习大量的知识，使自己能在短时间内，不管是在知识上还是在写作水平上有了明显的提高。

感谢 SQL Server 圈子中的很多高手，特别是 MVP 宋沄剑、张骞、林勇桦、王成辉、陈畅亮、蔡传雄，还有某电商 DBA 侯亚俊，某公司的 CIO 矫正，以及这个仓促写就的名单之外的更多朋友们，感谢你们长期的支持和帮助。

谨以此书，献给我最亲爱的家人，以及众多热爱 SQL Server 的朋友们！

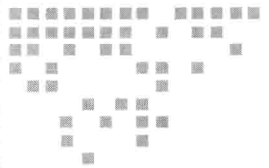
SQL Server MVP 黄钊吉

目 录 Contents

第一部分 *Part 1*

SQL Server 性能优化概述

本部分包含两章，主要介绍何为性能及相关的一些术语，因为只有了解什么是性能，才能开始进一步的学习。如果读者对此已经有了一定的基础和经验，可以跳过这部分，从第3章开始看起。但是基于连贯性，建议读者抽空看看这部分的内容。



Chapter 1 第 1 章

性能概述

在工作、论坛、博客及一些 QQ 群中，很多人总会问：怎样进行 SQL Server 性能优化？这个问题很难回答，一个千余人参与、发展了十几年的产品，其所涉及的性能优化并不是三言两语就可以说清楚的。想要熟练掌握该技能，需要系统地了解相关知识，而本书的主要目的就是帮助读者全面深入地把握知识结构。书中会告诉读者从哪里着手，优化的前提有哪些。但是，本书不是写给那些没有任何基础，连一个相对简单的查询语句都不会，或者连 SSMS（SQL Server Management Studio）都不知道怎么打开的人看的。如果你是这样的人，建议你先看其他入门书籍。

本章首先会对全书进行一个简单概述，帮助读者了解一些与性能相关的知识。因为在处理性能问题之前，我们首先要知道，什么是性能问题。

1.1 何为性能

想要进行性能优化，首先必须要了解性能问题，也就是说，最起码要对性能问题有一个较为明确的定义。试想，你生病了，去看医生，一到医院，你就对医生说：医生，我很不舒服，赶紧给我开药。医生连你哪里不舒服、怎么不舒服都不知道，凭什么开药呢？所以我们首先要知道面临的是什么问题，才能找到相应的对策。

作为 DBA，经常会被程序员、公司管理层问道：为什么数据库运行那么慢？如果你进一步询问他们慢到什么程度，有什么表征时，得到的答案往往又很模糊，可能只有慢或者卡。

所以在处理性能问题时，首先要对其有一个清晰的定义，不然会浪费很多时间去查找问题的所在。但是作为非专业人员，的确很难清晰定义所有的问题。为此，不妨来对性能

问题下一个非官方的定义：

在现有资源没有达到最大吞吐量的前提下，若系统（包括操作系统、数据库管理系统、应用程序等）不能满足合理的预期表现，则可以定义为有性能问题。

注意上面的限定词——合理。你不应该对所有的应用和操作都赋予很高的期望，比如，对于 OLAP 系统，它的某些操作往往需要大量时间和资源（比如 ETL），你不要期望它总是能在几秒内完成。当然，如果时间过长，也是可以定义为存在性能问题的。

另外，可以考虑一些非正式的定义，比如资源耗费明显过多、运行速度的下降超过规划速度等。

总之，在处理性能问题之前，尽可能给出清晰准确的定义，可以提高问题的解决效率。要分清什么是性能问题，什么不是性能问题（比如权限、某些硬件故障、某些程序 bug），因为不同的问题对应的解决方案往往是不同的，所以首先应该对性能问题给出定义。

1.2 性能指标

定义性能问题时往往可以有很多指标，其中最常见、最重要的指标有 3 个：响应时间、吞吐量、可扩展性。**响应时间**这个指标其实很明显，一个查询运行得快，性能问题通常很少，但如果某个查询的运行时间明显过长，那就说明可能有一定的性能问题了，需要引起注意。终端用户基本上只会关心他 / 她的请求是否能足够快地得到响应，所以他们的“性能”汇报往往只是告诉你它很“慢”，而这个“慢”其实指的就是响应时间。**吞吐量**可以理解为网络、设备、端口、虚电路或其他设备单位时间内成功地传送数据的数量，也可以理解为资源的使用情况。比如磁盘，每秒的吞吐量越大，传输的数据就越多，SQL Server 在向磁盘读写数据时延时就越短。**可扩展性**表示在遇到性能问题时，是否可以通过简单的增加资源的方法来解决问题。

对于性能指标，并没有一个固定值或者建议值，通常在要定义工作负荷（也叫做性能基线，将在后面章节中介绍）之后，通过监控及对比来把握。对于性能问题或者管理问题，常规的做法是先进行监控，然后分析监控数据，再根据分析结果进行处理，最后再次监控，一直如此循环往复，直到满意为止。

1.3 性能目标

打个比方，笔者比较崇尚中医的治疗方法，他们所采用的“望闻问切”诊断方法对准确掌握病人的病情很有帮助。记得有人说过，DBA 很多方面就像中医，在面对性能问题的时候，使用这种中医的诊断思维去处理会事半功倍。其实就是通过多方面检测，找到性能问题的根源和一些潜在风险。中医的一个思想是通过调理人体自身机能去抵抗外部的入侵，

旨在把人体调整到一个“平衡”状态，而不是像西医那样直接杀死染病细胞或者病毒。个人认为，优化性能的目的也是把系统调整到平衡状态，要把事情做到极致，但是不要极端。比如，不要花几个小时的时间，去尝试优化一个已经在 1 ～ 5s 内能得到结果的查询。

对于 SQL Server 的优化，一个比较通用的目标就是：尽可能最小化每个 SQL 语句（或者请求）的响应时间并增加系统的吞吐量，通过减少网络延时、优化磁盘 I/O 吞吐量以及减少 CPU 的处理时间来最大化整个数据库服务器的伸缩性，使系统能够协调运作。

简而言之，性能优化的目标就是通过一系列的手段，使系统能够协调、平衡地运作，合理地响应外部及内部请求，实现资源利用的最大化。

1.4 影响性能的常见因素

在面对性能问题的时候，我们的眼光和思路不能只局限在 SQL Server 中，因为很多因素都有可能导致出现性能问题。常见的影响性能的因素主要包括以下几方面。

- 应用程序的体系结构
- 应用程序设计
- 事务和隔离级别
- T-SQL 代码
- 硬件资源
- SQL Server 配置

1.4.1 应用程序的体系结构

这部分内容有专门的程序优化书籍，本书不会讲述太多。应用程序的体系结构如果不合理，将会导致数据传输的速度和开销增大，特别是网络层面和并发数。很多年前，应用程序三层架构已经风行全球，典型的三层架构如图 1-1 所示。

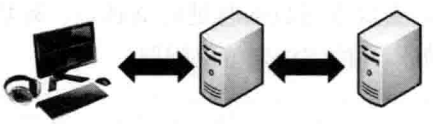


图 1-1 应用程序三层架构

这三个层次是一个整体，任何一部分出现性能瓶颈都将导致系统性能下降，而且会被用户明显地感觉到。

第一层是客户端，它可以是一个网页或者其他应用程序。这部分的问题常表现为网络带宽和客户端机器性能不足（比如网页程序上加载一个庞大的数据集）。

第二层为中间层或者应用层，主要处理事务逻辑。除此以外，它还可以起到杠杆作用，向应用程序提供更好的性能和扩展性，比如通过缓存常见用户请求结果以最小化重复计算、通过在这层使用多个服务器以分摊工作负荷、通过共享一个 SQL Server 连接以最小化会话数等。SQL Server 的每个连接大概需要 50KB 的存储空间，所以减少不必要的连接可以减轻资源需求方面的压力。

第三层是数据层，主要处理应用程序发出的数据请求，比如存储、查找、操作数据等。本书的主要研究对象就是这一层。

1.4.2 应用程序设计

笔者见过很多性能问题，实际上都不是SQL Server的问题，比如从前端导入数据时，部分开发人员在导入的过程中使用前端语言过多地处理数据逻辑，把一些数据库强项放到前端处理，往往就导致应用程序端的内存溢出等问题。

还有一种比较常见的情况，就是前端程序用游标去读取大数据量的结果集，这一方面会导致速度变慢，另外一方面会使得数据集占用内存过久，影响其他请求的性能。在撰写本书过程中笔者还遇到过一种情况：编程语言是Java和dorado，由于展示方式的错误，导致登录应用程序之后一直没有响应，后来发现是首页的树结构问题，树结构菜单展开后会增加SQL服务器的压力。针对问题修改之后运行恢复正常。

应用程序的设计对性能也非常重要，如何处理事务的逻辑、工作流是否合理，以及安全性、会话管理和缓存机制等都对性能和扩展性有很重要的影响。请记住，应用程序和SQL Server组合出来的系统是一个整体，不能单独讨论，但是本书不是百科全书，所以只能针对一些特定的部分进行讨论。除了前端程序，还有数据库的设计，这部分也是很专业的知识点，但它超出了本书范围。

1.4.3 事务和隔离级别

这部分是最常见的性能问题的根源之一，将在第9章中展开描述。任何成熟的数据库管理系统都应该支持多并发及保证数据一致性，SQL Server通过事务及锁来实现这方面的需求。与SQL Server交互的应用程序需要用到一个或多个事务，这些事务可以显式或者隐式地启动。作为应用程序设计者，首先应该注意最小化阻塞问题，隔离级别为应用程序提供了一个选择，是选择一致性还是并发性？一致性越强，并发性就越差。另外，需要控制事务运行的时间，这个时间决定了独占锁的生命周期，比如有些锁会一直持续到事务结束才被释放。

1.4.4 T-SQL代码

T-SQL是操作数据库的工具之一。大部分的性能问题都是编码引起的，所以编写高效、可维护的代码，对性能的影响不可估量。在面对大数据集时，选用面向集合的T-SQL操作数据库比使用面向过程的游标处理，效率上也高出很多个数量级。另外，除了性能上的考虑，也要考虑T-SQL编码规范，如果拥有编码良好的T-SQL，即使出现性能问题，也能很快地定位。本人经常要对一些两三千行没有注释甚至注释是错误的存储过程代码进行优化，这种工作非常辛苦。