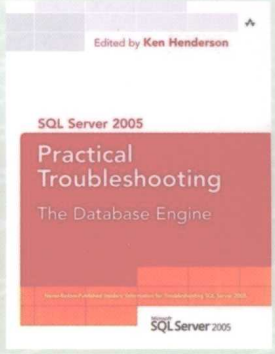
TURING 图灵程序设计丛书 数据库系列

Addison Wesley

SQL Server 求生秘籍

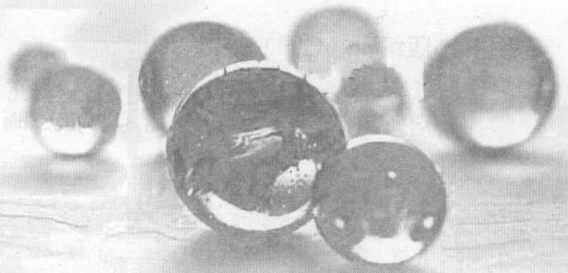
[美] Ken Henderson 主编
微软SQL Server 开发小组和支持部门 著
若启 一辉 瞿杰 译

- 微软SQL Server内部技术资料大曝光
- 来自SQL Server开发小组和支持部门的梦之队打造
- SQL Server故障排除圣经



人民邮电出版社
POSTS & TELECOM PRESS

TURING 图灵程序设计丛书 数据库系列



SQL Server 求生秘籍

[美] Ken Henderson 主编
微软SQL Server开发小组和支持部门 著
若启 一辉 瞿杰 译

人民邮电出版社
北 京

图书在版编目（CIP）数据

SQL Server 求生秘籍 /（美）亨德森（Henderson, K.）主编；若启，一辉，瞿杰译．—北京：人民邮电出版社，2009.2
（图灵程序设计丛书）
ISBN 978-7-115-19111-3

Ⅰ. S… Ⅱ. ①亨…②若…③一…④瞿… 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（2008）第169280号

内 容 提 要

本书帮助你解决众多数据库引擎方面的问题，每一章从关键的 SQL Server 组件入手，然后探讨用户遇见的常见问题，并给出解决方案。本书的主要内容包括等待和阻塞、数据毁坏和恢复、内存、过程缓存、查询进程等。本书的作者都是来自微软公司 SQL Server 开发团队和客户支持服务部门的支持专家。在你的 SQL Server 系统遇到问题时，本书将变得不可或缺。

本书适合数据库管理员和数据库开发人员阅读。

图灵程序设计丛书

SQL Server 求生秘籍

◆ 主编 [美] Ken Henderson
著 微软SQL Server开发小组和支持部门
译 若启 一辉 瞿杰
责任编辑 傅志红
执行编辑 陈兴璐

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号
邮编 100061 电子函件 315@ptpress.com.cn
网址 http://www.ptpress.com.cn
北京顺义振华印刷厂印刷

◆ 开本：800×1000 1/16
印张：22.25
字数：554千字 2009年2月第1版
印数：1－3 000册 2009年2月北京第1次印刷

著作权合同登记号 图字：01-2007-3151号

ISBN 978-7-115-19111-3/TP

定价：59.00元

读者服务热线：(010)88593802 印装质量热线：(010)67129223

反盗版热线：(010)67171154

版 权 声 明

作者简介

本书的创作团队由来自SQL Server开发小组的开发人员以及来自微软的客户支持服务机构的支持专家组成。其中7位来自SQL Server开发小组，3位来自微软的客户支持服务部门。

SQL Server 开发小组

August Hill是一位具有30多年开发经验的开发人员。在过去的6年里，他一直是SQL Server Service Broker小组中的一员。他为该产品在可支持性方面做出了许多贡献。在业余时间里他喜欢弹吉他以及品味华盛顿葡萄酒。他的联系方式是august.hill@microsoft.com。

Cesar Galindo-Legaria是SQL Server查询优化器部门的主管。他于1992年获得哈佛大学计算机科学专业（数据库方向）的博士学位。在波士顿的一家图形公司工作过一段时间以后，他又返回到数据库领域，在欧洲研究中心进行博士后访问。他于1995年加入微软从事新型关系型查询处理器的工作，该处理器随SQL Server 7.0一起首次发布，其中引入了一个完全基于开销的查询优化器、一套丰富的执行算法以及许多自动管理的特性。从那时起，他就一直从事SQL Server查询处理的工作。他在查询处理和优化领域拥有多项专利，并且在该领域内发表了许多研究论文。

Ken Henderson是一位具有25年以上开发经验的开发人员。他从1990年开始就从事SQL Server的工作并在其职业生涯内为多家公司开发软件，包括H&R Block、美国中央情报局、美国海军、美国空军、Borland国际公司、摩根大通（JP Morgan）等公司。他于2001年加入微软，目前是SQL Server开发小组中易管理性平台部门（Manageability Platform Group）的一位开发人员。他是SQL Server 2005中SQLDiag工具的开发者，并专注于SQL Server管理工具及相关技术的工作。他是8本涉及多种计算机主题的作者，包括出版自Addison-Wesley公司的SQL Server畅销图书*Guru's Guide*系列。他和全家住在达拉斯，他的电子邮箱是khen@khen.com。

Sameer Tejani出生于坦桑尼亚阿鲁沙，在过去的10年里工作于微软的SQL Server部门。他的工作经历使他能够接触到SQL Server引擎的不同领域，包括T-SQL执行框架、开放数据服务（ODS）、连接管理、用户模式调度器（UMS）以及其他领域。他就是那个令支持专家们痛恨的臭名昭著的“non-yielding scheduler”错误消息的唯一负责人！他目前是SQL Server安全小组的软件开发主管。在业余时间内，Sameer喜欢户外活动和长途骑车。他和妻子Farhat住在西雅图。

Santeri Voutilainen，更为常用的名字是Santtu，自从1999年以来就在SQL Server存储引擎小组担任软件设计工程师。他的工作与页分配、闩锁以及锁管理器密切相关。他毕业于哈佛大学，并在华盛顿大学获得计算机科学的硕士学位。尽管称西雅图为自己的家乡，但Santeri生于芬兰，并在尼泊尔渡过了大部分的青年时光。他非常热衷于旅游和户外运动，在闲暇时间内携其妻子与一

岁大的儿子在西北太平洋地区进行探险。Santtu的联系方式是santtu@vode.net。

Slava Oks是SQL Server的存储引擎和基础结构小组的软件架构师。他在微软渡过了9年多的时光。在SQL Server 2005的开发项目中，他负责SQLOS的架构和实现。他为该产品在性能、可扩展性、可支持性以及可测试性方面做出了许多贡献。他也是SQL Server的热门博客blogs.msdn.com/slavao的作者。他喜欢在业余时间进行体育运动或者与家人和朋友们一起娱乐。

Wei Xiao从1996年—2006年4月在微软负责SQL Server存储引擎的设计。他主要专注于访问方式、并发模型、空间管理、日志记录以及恢复机制。他也曾从事过SQL Server性能监控和故障排查的工作。他在几次行业会议中做过演讲，包括微软技术大会（TechEd）以及SQL PASS。他目前正在从事微软内部的一个数据存储项目的工作。

微软客户支持服务

Bart Duncan已经从事SQL Server及相关技术有大约10年时间了。他目前是SQL Server产品支持部门的专家级工程师。Bart与他的妻子Andrea Freeman Duncan医生居住在得克萨斯州达拉斯市。

Bob Ward是在位于得克萨斯州艾文市的微软区域支持中心的微软客户服务和支持部门工作的资深专家级工程师。他在微软服务了13年，并且迄今为止已经支持过微软SQL Server的每一个发行版本，从OS/2平台下的1.1版到SQL Server 2005。他在计算机行业的背景持续了20年，包括像在通用动力（General Dynamics）、Harris Hospital以及美国航空公司这样的公司中从事数据库开发项目。Bob于1986年在贝勒大学获得了计算机科学的学士学位。他目前和妻子Ginger以及两个儿子Troy和Ryan居住在得克萨斯州North Richland Hills市。Bob在业余时间会把精力放在青少年体育辅导，支持本地职业体育运动队，并努力提高自己的高尔夫球水平以实现参加PGA巡回赛的梦想。

Cindy Gross从2000年开始成为得州微软PSS的SQL Server和Analysis Services支持小组的成员。在此期间，她担任过许多角色，包括支持工程师、内容主管以及Yukon readiness主管。在加入微软以前，Cindy曾经是一位具有7年经验的SQL Server DBA，负责SQL Server 1.11版及其后续版本的工作。她还是一位热忱的科幻小说读者，关注女性独立的相关图书。她最喜欢的非技术作家是Sheri S. Tepper。Cindy在周末经常开着她的越野摩托（目前是一辆2004年款的本田CRF250X）赛车。你可以从Cindy的网站联系到她：http://cindygross.spaces.live.com/。

前　言

原本我想在本书中让微软技术支持工程师撰写多年来在SQL Server的技术支持工作中所学到的知识。当我加入微软后，令我惊奇的是，技术支持工程师们并没有把关于产品支持的实践知识（在认识论中叫作“领域知识”）记录下来。这些知识仅仅停留在口口相传的状态。

当然，这导致了一个问题：人们并不知道如何做好工作，除非有热心人来向他们展示该如何做。这也是一种非常容易犯错误的方式，会导致一些最重要的产品支持知识集中在少数人手中——这些经验被他们充分利用，但其他的支持团队却不了解这些知识。

加入微软之前，我做全职软件开发工程师已经20多年了。令我十分惊讶的是，原来支持部门的高端人群都是些曾经做过开发的人。通常，在成为技术支持工程师之前，他们都有3～5年的开发或相关工作经验。作为一名职业开发人员，我很难想象技术支持也可以做长工。对于我来说，支持工作似乎与软件开发世界中的看门人类似。他们不得不帮那些编写乱糟糟代码的开发者“擦屁股”的人。虽然我知道这很重要，但是私下里还是觉得把技术支持工作作为职业并不是一件开心的事。尽管如此，确实有好几个程序员前辈呆在技术支持部门，这让我感到迷惑。

于是，我开始思考如何创造人人机会均等的局面，让原本只有技术支持的上层梯队才能拥有的知识可以传达给每个人。奥林匹斯山上的家伙们对拥有产品支持的全面知识和领域知识引以为豪，对于我来说，这些知识应该分享给组织中的每个人。每一个做产品支持的人都应该有相同的权限来了解它们。

我最初打算在我写的*The Guru's Guide to SQL Server Architecture and Internals*一书中加入如何对产品进行故障排查的内容。然而，我很快意识到，开发软件或从开发人员的角度理解软件与故障排查有本质上的不同。可以说，它们是两种完全不同的领域知识。虽然有些部分是重叠的，但对产品故障排查而言，肯定有一些属于它自己特有的东西。

在我最终完成那本关于架构的书后，开始回过头来思考这个问题——如何把支持组在多年里学到的关于产品故障排查的许多低层次的细节和洞察记录下来，并不会有很多关于产品如何运行的详细信息，而是如何解决与产品有关的问题的详细信息。于是我开始与一些支持工程师探讨这个想法，并试探他们对此的兴趣。我建议做一个多作者的项目，在该项目中他们可以把一些难得的故障排查经验发表出来——这不仅仅是为了那些做技术支持的同事们，也为了客户。许多东西还从未被出版过，我感到，如果他们可以看到自己的文字出现在出版物中，最终会激励他们投身于写一些可能只有微软内部的少数人知道的东西。

什么样的回应都有，从不太热情到很热情。在翻遍了关于哪些人感兴趣和哪些人不感兴趣的花名册之后，我发现，很明显，我需要更多的作者加入到本书的写作队伍中来。在支持部门中，

愿意并能够投身到该项目中的人寥寥无几。

我本可以抛开支持组，而跑到微软咨询服务部门，或干脆去找那些最有价值专家（MVP）和类似的人选，但其实我很想把作者队伍限制在那些看过SQL Server源代码的人之中。访问过源代码并一步一步调试过，这种经历是不可替代的。通过研究产品代码，可以更深地理解SQL Server技术，从一定意义上讲，其他方法是做不到这一点的。

因此，我们还需要多引入一些作者，我决定向产品组的一些顶级开发人员发出邀请。虽然微软的产品组与支持组截然不同，他们关注的是开发产品，而不是提供技术支持，但我私下认识他们中的许多人，并知道他们已经花了大量的时间来调试和解决产品中的一些复杂问题，特别是对他们自己所创建的那部分。如果你从不调试代码，是写不出复杂代码的。我相信他们肯定会觉得新颖且有趣，愿意为本书增加一些实际的故障排查知识。

产品组的同事给我的答复都很热情，有好几个来自SQL Server组的顶级开发人员同意参加这个写作项目。我终于成功邀请到了真正写代码的人来探讨如何进行复杂且常见的问题的故障排查。在其他书中，你是找不到这些的，作为参与者，能够让你看到这些内容让我感到激动。[①]

我那本讲架构的书告诉你SQL Server是怎样运转的，这本书则告诉你SQL Server要是不转了怎么办。前一本书，不管你要处理SQL Server的哪一块，都用得上。而现在这本，按说只用于极端情形（因为SQL Server这个产品并不会经常挂掉），然而你的SQL Server应用是一直能让用户满意呢，还是会随时引起用户的怒火，有没有本书差别可就大喽。当然我希望你不会在使用SQL Server时遇到问题。如果遇到了，本书将是你开始故障排查征程的起点。

在此，我要感谢SQL Server产品组中那些做开发的同事们，他们为本书提供了不少内容，分别是：August Hill, Cesar Galindo-Legaria, Sameer Tejani, Santeri (Santtu)Voutilainen, Slava Oks和Wei Xiao；我也要感谢几位技术支持工程师，是他们为这个项目提出了宝贵意见：Bart Duncan, Bob Ward和Cindy Gross。他们都有自己独特的思考（和写作！）方式，但帮助你处理SQL Server故障排查中的一些实际问题，这帮人再合适不过了。

Ken Henderson[②]

① 随书附赠的部分章节代码请于图灵教育网站（www.turingbook.com）上下载。——编者注

② Ken Henderson已于2008年1月27日去世，享年41岁。——编者注

致　谢

Cindy Gross：我想要感谢Ken Henderson邀请我参与到本书的创作中来。这是一段美妙的经历！我要感谢每位PSS SQL Server数据服务小组的成员，你们所有人帮助我掌握群集技术的窍门。当然，如果离开了我丈夫Pat Burroughs的支持也是无法完成这一切的。Pat总是不顾长时间的工作以及充满陌生缩写词的电话会谈，给予我强大的支持。最后但仍是重要的，感谢我们的猫咪Golgothan嚼烂了电源线，从而促使我们购买了一台新的笔记本电脑。还要感谢我们的另一只猫咪Serendipity，让我们每天挠她的背十几次。

Santeri Voutilainen：我想要感谢Ken Henderson把本书的各章节集中在一起。也要感谢我的妻子给予的爱和鼓励，并一直温和地督促我进行写作。还要感谢我的儿子带给我的欢乐，这是一种只有一岁大的快乐的小孩才能带来的欢乐。

Bob Ward：我要感谢Ken Henderson提出这个想法，感谢他将其推荐给Addison-Wesley，并感谢他在临近截止期时，极其耐心地一遍一遍地提醒我们所有人。

我还要感谢我的全家。首先，感谢我的父母Don和Annette Gibaud，他们具有强烈的责任心并一直引我向善。第二，要感谢我的妻子Ginger以及我的儿子Troy和Ryan，感谢你们包容那些忙碌的深夜，从而使我能够完成本书的工作。如果没有你，Ginger，我不知道我该如何才能正常地进行写作。我还要感谢主为我的生活带来了能够每天包容我的人。我将会一生一世珍惜你的恩赐。还有我的两个儿子，Troy和Ryan，你们是我的骄傲和欢乐。原谅我每次把你们拽到户外，训练你们的投球技巧以及投篮技术。我很希望你们能在自己非常喜欢的东西上能有所成就。请记住，我是一个问题解决者，而有时要把这顶帽子摘下来真不是那么容易啊。愿上帝祝福你们俩并能一直成为你们的努力、力量以及进取的源泉。

Bart Duncan：本书中涉及的我所了解的许多主题，应该是我在微软的同事们的共同财富，他们非常慷慨地将他们的时间和知识与我分享。能够在一个拥有大量杰出的人们并对其所从事的工作充满激情的小组里工作，我感到非常幸运。其中，Ken Henderson应该得到特别的赞誉，他提出了隐藏在本书背后的那些想法，并联系到具有相应领域的知识且能参与到该项目中的同人，而且还要感谢他在项目常常处于停滞状态时凸显出来的强大凝聚力。

当然，我无法用言语来表达对我的妻子Andi、我的父母Wendell和Lin Alyn Duncan、我的兄弟Lane以及我的儿子Walker和Texas Ranger的感激之情。多年来，他们以我无法言表的各种方式一直给予我强大的支持和鼓励。

目　录

第 1 章 等待和阻塞

Santeri Voutilainen

本章内容

- 等待类型
- 对阻塞问题进行故障排查
- 识别阻塞
- 确定阻塞的原因
- 资源类型的细节
- 死锁
- 监视阻塞
- 小结
- 其他资源

本章主要讲阻塞，但首先来讲讲等待。为什么要先讲等待呢？因为这两者具有非常紧密的联系，而且它们经常被当作同义词。因为它们联系紧密，与这两个词有关系的SQL Server概念和工具也容易被混淆，所以，对它们进行区分十分重要。

从概念上讲，等待通常涉及一个闲置状态，在该状态中任务会等待“一些东西”的触发，否则不会继续执行。这里的“一些东西”可能是同步资源的获取、新的命令批（command batch）的到达，或者其他事件。这个描述比较准确，但有一点需要注意：在SQL Server中，并不是所有处于闲置状态的任务都可以被标识为等待状态。这是因为等待的分类有时被用来表示执行一段特定类型的代码，这种代码通常不受SQL Server的直接控制。事实上，SQL Server的任务会等待外部代码执行完毕。

1.1 等待类型

SQL Server中的所有等待都被归类为不同的等待类型。一个任务的当前等待类型是根据等待原因来设置的。每个等待类型会被赋予一个名字来表示，如位置、组件、资源或等待原因。

尽管其中一部分名字不好理解（将在稍后讨论），但是另外一部分则可顾名思义。等待类型的列表可以通过`sys.dm_os_wait_stats`动态管理视图（dynamic management view，DMV）获得。默认情况下，DMV的输出是没有完整列表的。对于SQL Server 2005来说，SQL Server产品组倾向

于不包括以下3种等待类型。

- 等待类型从未在SQL Server 2005中使用过。注意，一些没有被排除的等待类型也从未被使用过。
- 等待类型仅在不影响用户行为时发生，例如在服务器启动或关闭的时候，它们对用户是不可见的。
- 由于较高的发生频率或者较长的持续时间而引起用户关注的但无害的等待类型。

不幸的是，这些等待类型的缺失会导致一些问题，因为最后一类等待类型确实在其他的等待源（也就是sys.dm_os_waiting_tasks DMV）中出现过。完整的等待列表可以通过启用跟踪标志8001来获得。这个标志唯一的效果就是强制sys.dm_os_wait_stats显示所有的等待类型。

```
dbcc traceon (8001, -1)
```

注解　该跟踪标志是未经文档记载的，类似于其他未经文档记载的特性一样，它没有得到支持，所以你必须自己承担使用它所带来的风险。

等待类型列表可以被划分为4个基本类别：资源、计时器/队列、IO操作和外部等待。资源等待是其中最大的一类。它涵盖了如同步对象、事件以及CPU等资源的等待。计时器/队列等待包括了等待计时器超时，或者等待处理队列中的新项目。IO等待包括了大多数涉及IO操作的等待。网络和磁盘的IO操作都包括在内。外部等待涉及之前提到的情况——任务正在执行某些类型的代码，对SQL Server来说通常是外部的，如扩展存储过程。

当前处于等待状态的任务列表可以从sys.dm_os_waiting_tasks中获得。关于等待，DMV包括了用于确定等待任务、等待的持续时间、等待类型以及在某些情况下关于正在等待什么的附加信息。DMV还包括了与阻塞相关的信息，也就是阻塞任务继续执行的进程的身份以及获悉该身份的时间。它也是阻塞信息的根源，因此了解如何使用DMV从普通的等待中区分出阻塞情况是非常重要的。

阻塞与非自愿等待不同，它是受到另一个阻止其worker执行的任务所迫而进行等待的。当这些任务试图同时访问某个共享资源[①]时，就会出现这种情况。通常，这个定义被认为是把外部型与计时器/队列型等待排除在阻塞中的等待任务之外的，尽管该定义并不严格排除外部等待。本章结尾包含了一些关于外部型与计时器/队列型等待的内容。

1.2　对阻塞问题进行故障排查

研究和处理阻塞有3个通用的步骤。

(1) 识别已发生的阻塞。

(2) 确定导致阻塞的原因。

(3) 消除导致阻塞的原因。

本章主要关注前两个步骤。由于阻塞可能出现在各种组件中，且阻塞的原因跟那些组件相关，

① 通常，这需要有两个任务。然而，一个任务也有可能阻塞它自己。在现实中，这种情况很少发生，而且通常会造成死锁。这种情况将影响到用于检测或者研究阻塞的脚本。

因此解决方案也常常是组件特性。可以参考关于那些组件的文档。本章包含了一些常见阻塞类型的原因及解决方案。

注意，这里的大多数实例和讨论都是基于SQL Server 2005 SP1（或者后续版本）的，尤其是在使用sys.dm_os_waiting_tasks DMV时。SP1中包含了对该DMV的一些改动和修正。SQL Server 2005 RTM版的行为与接下来的描述有明显不同。

1.3 识别阻塞

1.3.1 通过 sys.dm_os_waiting_tasks 来识别阻塞

在SQL Server 2005中，sys.dm_os_waiting_tasks DMV是服务器中主要存储阻塞信息的地方。在讨论如何用其来检测阻塞的存在之前，先来看看其中一些比较重要的列。

sys.dm_os_waiting_tasks包括3组列：一组标识等待者、一组标识阻塞者（如果存在阻塞者）以及一组提供等待信息。

标识等待任务的列是waiting_task_address、session_id以及exec_context_id。waiting_task_address包含代表该任务的对象的内存地址，在SQL Server内唯一地标识一个任务。session_id代表当前与此任务相关的用户会话。任务与会话之间的关联只存在于任务持续期间，通常在一个批处理之内。阻塞的任务通过blocking_task_address、blocking_session_id和blocking_exec_context_id来标识，这些列与标识等待任务的列具有相同的语义。

要注意任务地址和会话ID列之间的关系。只有那些当前在SQL Server内正在执行语句的会话才有一个对应的任务。不过，一个任务不一定对应一个会话ID，各种系统任务都会遇到这种情况。由于SQL Server中的所有等待都与任务相关，因此当前没有相关任务的会话不可能处于等待或被阻塞状态。所以waiting_task_address永远不会为空，但如果该任务没有相关会话，那么session_id和exec_context_id可能为空。另一方面，任务可以被活动任务或者非活动会话所阻塞。如果等待中所涉及的资源被一个跨任务（也是跨批处理）的会话所持有，那么就有可能发生这种情况。在SQL Server 2005中，可能引起阻塞并且被跨批处理持有的资源只有锁。注意，尽管与会话相关的其他一些资源（比如内存）是跨批处理持有的，但它们并不会直接导致阻塞，而是通过代理等待（proxy wait）来表现（比如LOWFAIL_MEMMGR_QUEUE等待类型）。锁被事务所持有，并且由于一个事务可能包含多个批处理，从而可能出现这种情况：有一个活动事务的会话在执行客户端操作时仍然持有锁，因此该会话在服务器上就不会有相应的任务。

处于阻塞状态的任务的标识并不总是已知的。在这种情况下，标识阻塞任务的那些列就为空。这是很普遍的，因为大多数等待类型都没有可用的阻塞者信息，要么是因为它对于这个特定的类型没有意义，要么是由于没有捕捉到相关的信息（比如出于性能考虑）。

顾名思义，wait_type列包含了处于等待状态任务的当前等待类型。与此类似，wait_duration_ms列包含了当前等待的持续时间。resource_address列提供任务所等待资源的内存地址，它主要用作一个标识符，当等待类型名称相同时，可以区分在一个资源不同实例上的阻塞。在这种情况下，resource_description列也能提供区分信息，但它只对少量的等待类型提供有用信息——这些类型包括所有的锁和闩锁等待类型、CXPACKET以及THREADPOOL等待类型。

使用sys.dm_os_waiting_tasks来检测阻塞最简单的方法是运行T-SQL查询select * from sys.dm_os_waiting_tasks，并将输出的任何记录都作为阻塞的征兆。然而这样做并不是很有效，因为几乎一直有一些任务等待属于那些通常可以排除在阻塞研究之外的类型。比如，死锁监视器线程常常被显示处于等待类型为REQUEST_FOR_DEADLOCK_SEARCH的等待状态，这仅仅说明了它正处于死锁检测事件间的暂停状态，并不是被阻塞。在一种情况下不应忽略这类等待，即如果处于这些等待状态的任务正在阻塞别的任务，只有WAIT FOR等待类型才可能出现这种情况。

尽管要记住那些可排除的等待类型并把它们从结果集中排除是可能的，但这样做通常效率不高。在sys.dm_os_waiting_tasks上建立一个视图有助于此。

在随书附赠的CD[①]中可以找到amalgam.innocuous_wait_types视图，它产生的是那些通常可以忽略的无害等待类型的清单。下面这个amalgam.dm_os_waiting_tasks_filtered视图可通过该帮助视图将这些无害的等待过滤掉。

```
create view amalgam.dm_os_waiting_tasks_filtered
as
select *
from sys.dm_os_waiting_tasks
where wait_type not in (
      select *
      from amalgam.innocuous_wait_types)
go
```

过滤后的记录集中剩下的任何记录都代表了真正被阻塞的任务。可以基于阻塞的严重程度、原因以及性质来进一步分析和缩小这个集合。在确定阻塞原因的小节中，你可以学到更多关于这种分析的知识。

为什么不用*sysprocesses*或*sys.dm_exec_requests*

在SQL Server的早期版本中，研究等待和阻塞所选择的DMV是sysprocesses。尽管在SQL Server 2005中这个DMV仍然存在，但已经不推荐使用了，而是用一组新的DMV来代替。包含sysprocesses中的等待及阻塞信息的新DMV是sys.dm_exec_requests。尽管这些DMV都包含可用于研究阻塞及等待问题的信息，但我们有很好的理由不再使用它们。

不应该使用这些视图的主要原因是，它们不提供像在sys.dm_os_waiting_tasks中那么详细的信息。sysprocesses和sys.dm_exec_requests显示的是会话级别的信息。这意味着它们不包含那些与会话ID不相关的系统进程信息。它们的基于会话也使得处理并行查询（相同会话ID下执行多个任务）更为困难。当研究涉及并行查询的阻塞时，sys.dm_exec_requests没有用处，因为它只显示父任务的阻塞/等待状态，而不显示子任务。

关注会话级别也影响到阻塞者信息的显示。sysprocesses和sys.dm_os_waiting_tasks都只显示处于阻塞状态的任务的会话ID，而不是并行查询中的特定任务。

最后，sysprocesses和sys.dm_exec_requests中对资源的描述不如sys.dm_os_waiting_tasks中的全面，后者提供了最完整的资源描述集。此外，sys.dm_os_waiting_tasks中的resource_address列可用于区分没有相关描述的多个资源，但通过其他两个DMV不能够对这种多资源进行区分。

① CD内容可以在图灵教育网站www.turingbook.com上下载。后同。——编者注

尽管sys.dm_exec_requests不应该被用作阻塞信息的主要来源，但它确实包含了一些调查特定阻塞类型时有用的信息，因此不应该忽略它。

1.3.2 从统计上识别阻塞

虽然sys.dm_os_waiting_task是个有用的资源，但它并不是检测阻塞的唯一资源。关于等待和阻塞的统计信息有几个来源。这种信息有时比通过sys.dm_os_waiting_tasks得到的当前等待及阻塞数据更有用，因为统计信息可以区分偶然的和频繁的，以及短暂的和长时间的等待。频繁的长时间阻塞通常比偶然的短暂阻塞更有意义。

除了提供前一节所提到的等待类型列表以外，sys.dm_os_wait_stats还提供每种等待类型的统计信息。这些信息包括某个给定等待类型的等待出现的次数、等待的总持续时间以及单次等待的最长等待时间。

也可以对这个DMV进行单个查询来找出那些平均等待时间长的等待类型。记住，只看该视图的一个快照所得到的绝对等待计数是没有意义的，因为这些计数是从上次重启之后一直累加的，而重启可能发生在以前的某个时刻。衡量一个特定时间段内的计数才有意义。两个快照之间的差别显示了在那个间隔之内的阻塞的严重程度。可以对这个DMV执行两次手动查询并计算差别来生成这些快照，更简便的方法是使用SQL Server 2005自带的SQLDiag工具。

```
-- Create temporary tables to store the initial
-- and final snapshots
--
create table #StatsInitial (
     wait_type sysname,
     waiting_tasks_count bigint,
     wait_time_ms bigint,
     signal_wait_time_ms bigint);
create table #StatsFinal (
             wait_type sysname,
             waiting_tasks_count bigint,
       wait_time_ms bigint,
       signal_wait_time_ms bigint);
-- Create indexes for join performance
--
create index idxInitialWaitType
       on #StatsInitial (wait_type);
create index idxFinalWaitType
       on #StatsFinal (wait_type);

-- Create an initial snapshot
--
insert into #StatsInitial
select wait_type,
       waiting_tasks_count,
       wait_time_ms,
       signal_wait_time_ms
from amalgam.dm_os_wait_stats_filtered;

-- Wait for a ten second delay
-- This delay can be adjusted to suit your needs
-- and preferences
--
waitfor delay '00:00:10';
```