

Oracle

Applications

性能调整手册

Oracle
Applications Performance
Tuning Handbook

Oracle Press™ 授权出版

(美) Andy Tremayne 著

刘谦 苏建平 等译



机械工业出版社
China Machine Press

OSBORNE Education

Oracle 技术系列丛书

Oracle Applications 性能调整手册

(美) Andy Tremayne 著

刘 谦 苏建平 等译



机械工业出版社
China Machine Press

本书是Oracle公司官方推荐的优化Oracle Applications性能的专著。本书全面系统地从Applications涉及的各个方面,其中包括客户、应用、网络、服务器等领域,向读者提供了端到端的全面解决方案。借助于本书提供的性能优化理论和实施方法,读者可以在解决性能问题的同时提高系统的容量,改善系统的可伸缩性,并延长各个系统组件的生存期。本书作者是Oracle Applications部门的权威人士,他总结了多年维护使用Oracle Applications的经验,提供了大量的真实优化案例供读者进行分析研究。本书附带的光盘提供了多种优化工具包。

Andy Tremayne: Oracle Applications Performance Tuning Handbook(ISBN 0-07-212549-7).

Copyright © 2000 by the McGraw-Hill Companies, Inc.

Authorized translation from the English language edition published by McGraw-Hill, Inc.

All rights reserved. For sale in the People's Republic of China.

本书中文简体字版由机械工业出版社和美国麦格劳-希尔国际公司合作出版。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。

版权所有,侵权必究。

本书版权登记号: 图字: 01-2000-3396

图书在版编目(CIP)数据

Oracle Applications性能调整手册 / (美)崔美恩 (Tremayne, A.) 著; 刘谦等译. -北京: 机械工业出版社, 2001.10

(Oracle技术系列丛书)

书名原文: Oracle Applications Performance Tuning Handbook

ISBN 7-111-09285-6

I. O… II. ①崔… ②刘… III. 计算机管理系统-应用软件, Oracle Applications-技术手册 IV. C931.9-62

中国版本图书馆CIP数据核字 (2001) 第055716号

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: 张明安 张鸿斌

北京昌平第二印刷厂印刷·新华书店北京发行所发行

2001年10月第1版第1次印刷

787mm × 1092mm 1/16 · 38印张

印数: 0 001-5 000册

定价: 79.00元(附光盘)

凡购本书, 如有倒页、脱页、缺页, 由本社发行部调换

译者序

经过三个月的努力，我们怀着喜悦的心情向广大工作在Oracle Applications领域的专业技术人员和那些希望提高Oracle Applications性能或正在寻求解决Oracle Applications性能问题全面解决方案的人员推荐本书。虽然市场上可以买到很多有关Oracle服务器和开发工具的书籍，但深入讨论Oracle Applications性能优化的专著却很少见。到目前为止，Oracle Applications所涉及技术的深度和广度已经迫使技术人员在对Oracle Applications进行性能优化或故障排除的过程中必须同时从多个技术领域中获取帮助。本书是目前市场上惟一一本从优化角度综合多种最新技术和方法论述优化的专著。译者作为多年从事Applications程序设计的专业技术人员，深感本书确实填补了Oracle Applications性能优化方面的空白。

在翻译本书的过程中，译者感到本书具有科学性、系统性、实用性三大特点。本书的最大特点就在于全面地向读者介绍了用于性能优化的性能方法论，该方法论在总结了作者多年从事Oracle Applications优化的经验的基础上系统地描述了调查及解决性能问题的具体步骤和方法，并得到了Oracle公司技术开发部门的认可和推荐。读者按书中提供的调查表格和提问就可以全面系统地展开性能优化工作。该方法论的出现首次把完全根据个人经验对Applications性能进行优化的手工方式上升到理论阶段。本书的系统性体现在作者在内容安排方面做到了自给自足，本书根据Oracle Applications所涉及的全部领域，从优化的角度向读者全面介绍了优化Applications所涉及的各种技术的基本理论和技能。其内容覆盖了从最基本的SQL、PL/SQL优化到最前沿的网络优化和最实用的PC机性能优化等所有与Oracle Applications性能优化有关的领域。考虑到Windows操作系统的普及，本书特别向读者介绍了如何对Windows操作系统（包括Windows NT和Windows 2000）中出现的性能问题进行分析 and 优化的具体方法，其中包括如何分析处理最令技术人员棘手的通用保护错误（GPF）等具体问题。值得一提的是，本书作者Andy Tremayne在Oracle公司有近十年的工作经验，本书中提供的所有优化方案都以实际案例为背景，读者通过分析这些案例不仅可以验证本书性能方法论的有效性，而且还可以为读者从事的具体业务提供有益的启发和帮助。除此之外，本书为了使读者可以尽快掌握并Applications性能方法论技术，附录中为读者提供了13个性能调查表，读者可以根据自己从事的应用来利用这些表格为性能优化和故障排除提供全面高效的服务。最后需要告诉读者的是，本书虽然是面向Oracle Applications的专著，但其介绍的优化理论和方法适用于所有从事计算机应用的专业技术人员。

本书由刘谦、苏建平负责校审和统稿，参与本书翻译工作的其他人员还有王军平、刘丽云、李新、李江月、张君哲、钱云、刘城、田红。刘树人、张建军、龙晶、刘颖、刘新等参与了本书的校对和录入。由于本书内容较新，篇幅较多。再加上译者的时间和水平有限，在翻译过程中难免有疏漏和错误，敬请读者给予批评指正。

2001年4月

序 言

在当今技术日益发展的时代，竞争的压力和我们为性能问题所付出的代价都要求得到更多的信息。为了满足不断增长的市场需求，在Oracle开发部门管理人员提议并支持下，本书的发行作为Oracle Applications提供了专家级的性能优化建议。

虽然市场上已经出现了不少有关优化Oracle数据库和相关技术的出版物，但除了Oracle文档外，本书首次从Oracle Applications的角度向读者介绍优化技术。Oracle Applications所涉及的技术深度和广度意味着性能优化信息必须取自于更广泛的技术领域。

本书在提供了有关性能优化方法和理论基础的同时，还为读者提供了多种优化工具和技术，从而可以实际提高读者解决具有挑战性问题能力。本书通过介绍预测和避免性能问题的方法来使读者具有找出问题和采用正确方法解决问题的能力。

虽然解决性能问题需要很大的开销，但如果预防得当的话，性能问题也是可以避免的。随着技术的发展，过去那种被动地定期性能调查方式已经不适应目前的环境。目前最流行的方法是借助于专业知识来快速确认解决问题所需的资源并主动地对应用进行监视。基于上述原因，本书中提供的建议和指导方针可以使读者有效地预防问题的出现。除此之外，改善性能的同时就意味着系统的容量、系统的可伸缩性和系统组件的生存期也得到了提高。本书推荐的多种解决方案可以在提高系统吞吐量的同时满足客户有限的财务预算。

本书中讨论的优化策略和推荐的优化建议是紧密围绕着Oracle Applications展开的，其目的是为了读者掌握有别于一般数据库和网络优化技术的新型优化方法。借助于本书提供结构化的优化方法，读者可以解决实际应用遇到的大多数问题。除此之外，本书中提供的大多数信息都是最新的，读者可以在运用这些知识的过程中逐步体会到这些信息的价值。为了确保本书内容的实用性，在出版之前，本书各章的主要内容都得到了Oracle公司高级专家的认可。

知识是最宝贵的共同财富。因此与读者和需要帮助的单位和个人共享作者的知识就是本书的最大愿望。值得一提的是，本书作者Andy Tremayne在Oracle公司有近十年的工作经验，并在最近三年中一直与Oracle开发部门的应用性能专业委员会密切协作。他的最大愿望就是与读者共享在长期工作中获得的经验和能力。本书的出版无疑是朝着这种愿望迈出的新一步。

Cliff Godwin

Oracle公司应用技术部副总经理

前 言

虽然市场上可以买到很多有关Oracle服务器和开发工具的书籍，但深入讨论Oracle Applications的专著却很少见。到目前为止，Oracle Applications所涉及的技术深度和广度已经迫使我们在对Oracle应用进行性能优化或故障排除的过程中必须同时从多个技术领域中获得帮助。出版本书的目的就是为了迎合读者在上述方面的需要。

本书中提供了许多鲜为人知的优化信息。尽管本书是面向Oracle Applications优化的，但本书提供的知识和优化的方法可广泛用于所有基于Oracle数据库的系统，除此之外，本书中介绍的大多数优化技术和基本指导原则也不仅限于对应用性能的调整。例如，某个著名的PC机制造商目前正在使用本书中有关PC机优化一章介绍的技术来优化该公司销往世界各地的PC机。

除了向读者介绍最新的优化技术之外，编制本书的主要目的是力图在提供有关性能优化的基本理论和概念的同时全面向读者介绍实用的Oracle应用优化技能。本书是作者积多年维护使用Oracle应用的经验，是在成功地帮助多达250个客户实现Oracle应用性能最佳化的基础之上的经验总结。

虽然本书某些章节的内容涉及到非常专业的技术领域，但只要读者具备了这些技术领域的基本知识，本书就可以把读者从非专业人员变为性能优化的专家。

本书第一部分向读者介绍Oracle Applications性能方法论的基本概念，这种专门用于优化应用性能的基本理论是在多年研究基础上并结合数百例性能优化案例的具体实践经验发展得来的。本方法论具有的简单、易行、高效的特点也是在大量优化实践的基础上总结得到的。对于解决复杂的涉及多个技术领域的性能问题来说，虽然不能过分强调系统方法的作用，但本书介绍的性能方法论可以向读者提供处理任何性能问题的基本框架和指导原则。性能方法中的一部分内容涉及到整理安装应用程序的过程和描述所面临问题的文档。第2章中介绍的项目文档摘要（PDS）为读者提供了一种综合各系统所有相关性能指标的结构化方法。除了需要在开始调查前收集必要的信息外，PDS文档摘要还可以为参与调查的所有工作人员提供问题的焦点并为不同专业的技术人员提供交流的手段。

本书第二部分的开始讨论了Oracle Applications所涉及的各种技术领域和展开调查的基本方法。这一部分中有关优化客户PC机的内容是读者必读的。这是因为对PC机的优化不仅可以为Oracle Applications创造最佳的运行环境，而且这种经过优化的PC设备还可以作为测试工具来帮助找出应用部件中存在的问题。本部分有关通用保护错误（GPF）一章专门向读者介绍了如何分析处理影响PC机稳定性的GPF和FRM-999999错误的方法。

本书第二部分中对包括中间件、数据库和SQL语句、服务器及网络在内的所有Oracle Applications系统的主要部件都给予了详细的分析。这一部分的每一章都在介绍不同部件的内部工作原理的同时向用户提供了性能优化建议和工具。有关专门讨论SQL语句的一章特别适合于具有跟踪调试窗体或程序进程经验的读者。需要强调的是，这一章中向读者提供了许多首次发

表的内部信息。即使读者认为自己了解SQL Trace，这一章也是不能跳过的。

本书第三部分的重点是关于用户、Oracle工作流和并行管理器的优化方法。在这一部分的有关章节中给出了某些用于比较PC机配置的时间参数和可用来规划或检查性能水平的网络延迟参数。由于并行管理器和工作流是比较复杂的系统组件，因此除了讨论有关这些组件的故障排除技术外，还详细介绍了这些组件的工作原理和配置及优化的方法。

之所以强调优化用户的重要性是因为那些没有经过培训的用户有可能会给已经优化的系统造成严重的后果。因此教会这些用户如何有效地进行查询并定期对他们进行培训，将有助于维护并改善系统的性能。本部分中有专门的章节向读者介绍作者根据自己的工作经验总结的快速查询技术和建立描述可变字段的技巧。借助于本书提供的这些技术和技巧，我们相信所有的读者，不管是用户还是部门经理，甚至是技术开发人员，都可以通过减少系统负载和取消没有必要的定制操作来节约可观的开支。

本书的第四部分主要向读者提供并讲解应用案例，这一部分的目的就是通过引入实际案例来使读者对本书提供的基本概念和方法有进一步的理解。本书的最后一章不仅对全书的内容进行总结，而且介绍一些著名的案例让读者了解可能在优化过程中遇到的特殊问题。除此之外，这一部分还向读者特别交代了与Oracle技术支持打交道时需要注意的事项。

最后，我们建议读者可根据自己的具体情况来参阅本书的附录部分。附录部分的重点是总结了编制项目文档摘要时使用的框架，并解释了Oracle技术支持要求用户使用的某些文档标准。附录的最后向读者介绍了使用技术支持请求（TAR）时的注意事项。

虽然本书向读者提供了数量巨大的技术资料并可作为读者实施优化工作的指南和参考资料，但我们希望读者在阅读和应用本书的技术时，一定要牢记：

尽管优化是一门科学，但在实现优化的过程中离不开判断和直觉的帮助。

本书英文版书名：Oracle Applications Performance Tuning Handbook.

英文书书号：ISBN 0-07-212549-7

英文书出版社网址：www.osborne.com

目 录

译者序

序言

前言

第一部分 用于调查和编制应用 性能文档的工具

第1章 性能方法论	1
1.1 性能方法论的引入	1
1.1.1 读者必须是专家吗	2
1.1.2 了解用户需求	2
1.1.3 性能方法论使用的工具	2
1.2 掌握系统技术	3
1.2.1 三种类型的应用系统	3
1.2.2 自己动手发现问题	3
1.3 整体化方法	4
1.3.1 第1阶段: 信息收集	6
1.3.2 第2阶段: 问题调查和优先级确定	9
1.3.3 第3阶段: 调查与验证	14
1.3.4 第4阶段: 进行优化	20
1.3.5 技术堆栈测量	23
1.4 问题定义	24
1.4.1 建立一个清晰简明的问题定义	25
1.4.2 案例研究	26
1.4.3 问题的性质What	27
1.4.4 发生问题的位置Where	32
1.4.5 问题发生的时间When	36
1.4.6 问题涉及的范围Extent	39
1.4.7 优先级确定Priority	42
1.5 案例研究小结	43
1.6 调查清单	44
1.7 本章小结	44
第2章 编制系统性能文档	47

2.1 项目文档摘要大纲	47
2.1.1 项目文档的优点	48
2.1.2 文档结构	48
2.2 如何回答提问	50
2.2.1 标题页总结信息	50
2.2.2 第1部分: Oracle联系人	53
2.2.3 第2部分: 客户联系人	53
2.2.4 第3部分: 问题定义	54
2.2.5 第4部分: 系统环境	56
2.2.6 第5部分: Oracle环境	61
2.2.7 第6部分: 应用环境	66
2.2.8 第7部分: Web及Forms服务器环境	69
2.2.9 第8部分: 客户环境	71
2.2.10 第9部分: 网络	73
2.2.11 第10部分: 进程信息	76
2.2.12 第11部分: 其他系统信息	79
2.2.13 第12部分: 进一步信息	80
2.2.14 第13部分: 对硬件和环境的要求	80
2.3 本章小结	81

第二部分 优化应用技术栈

第3章 开始调查	83
3.1 Oracle应用体系结构	83
3.1.1 Oracle应用部件	83
3.1.2 Oracle应用技术栈	88
3.1.3 其他Oracle应用部件	95
3.1.4 物理层实现	96
3.2 比较时限	99
3.2.1 自己动手进行性能测试	99
3.2.2 使用优化的PC机进行测试	102
3.2.3 Oracle应用测试	103
3.3 与优化有关的社会工程学方面的问题	105

3.4 如何开始调查	105	4.6.3 Oracle应用的字体别名	136
3.4.1 用户和业务处理	107	4.7 优化磁盘I/O	136
3.4.2 检查客户端配置	107	4.7.1 文件系统	136
3.4.3 检查网络	108	4.7.2 消除存储碎片和文件排序	138
3.4.4 检查服务器和数据库	108	4.7.3 检查硬盘工作模式	139
3.4.5 复查网络	109	4.7.4 使用直接存储器存取	142
3.4.6 记录并公布发现的问题	109	4.7.5 优化转储文件	143
3.5 本章小结	109	4.7.6 删除磁盘压缩功能	148
第4章 优化客户: Windows 95/98	111	4.7.7 优化磁盘高速缓存	149
4.1 优化指南	111	4.7.8 清除文件系统	149
4.2 开始优化前的准备工作	112	4.8 实用程序	154
4.2.1 检查Windows操作系统的版本	112	4.8.1 工具Power Toys	155
4.2.2 验证文件系统的完整性	113	4.8.2 工具Tweak UI	155
4.2.3 备份系统	114	4.8.3 工具KernelToys	162
4.2.4 确保简单的引导过程	115	4.9 故障排除	163
4.3 优化BIOS	122	4.9.1 检查硬件冲突	163
4.3.1 操作模式	122	4.9.2 使用Windows 进行长期测试	167
4.3.2 键盘速率	123	4.9.3 Windows操作系统的启动问题	167
4.3.3 高速缓存	123	4.10 有关Windows操作系统的提示	167
4.3.4 ROM映射	124	4.10.1 把桌面加入到菜单中	168
4.3.5 软盘引导	124	4.10.2 创建级联菜单	169
4.3.6 未使用的IDE通道	125	4.10.3 在文件中查找文本	169
4.4 优化Windows桌面	125	4.10.4 给程序指定键顺序	171
4.4.1 删除屏幕保护器	125	4.10.5 安装可热拔插的SCSI设备	172
4.4.2 删除墙纸	126	4.10.6 重新分配计算机	172
4.4.3 升级显示驱动程序	127	4.10.7 修改系统标志	173
4.4.4 减少屏幕使用的色彩数量	127	4.10.8 调整Windows系统的闪屏	174
4.4.5 优化图形性能	128	4.10.9 使用多语言操作系统	175
4.4.6 删除活动光标	129	4.10.10 FAT-16和FAT-32的比较	176
4.4.7 删除快捷工具条	130	4.11 修补程序与热修复	178
4.4.8 优化数据录入速度	130	4.11.1 Windows 95下的修补程序与 热修复	179
4.5 优化Windows环境	131	4.11.2 Windows 98下的修补程序与 热修复	181
4.5.1 优化设备的主要任务	131	4.11.3 Office 97下的修补程序与热修复	182
4.5.2 优化注册表	133	4.11.4 Novell环境下Windows 95/98的客户 升级	182
4.6 优化存储器	134		
4.6.1 优化CD-ROM配置的存储器	134		
4.6.2 减少已安装的字体数量	135		

4.12 Windows操作系统的检查清单	183	5.7.5 有关使用二级高速缓存的考虑	234
4.13 本章小结	183	5.7.6 Windows NT 服务程序包和热修复	235
第5章 优化客户: Windows NT/2000	186	5.8 本章小结	235
5.1 优化指南	186	第6章 通用保护错误	239
5.2 优化前的准备工作	187	6.1 什么是GPF错误	239
5.2.1 Oracle应用对存储器和CPU处理器的 要求	187	6.1.1 GPF与MS-DOS服务的相关性	241
5.2.2 校验文件系统的完整性	187	6.1.2 可再现或随机发生的GPF错误	241
5.2.3 备份注册表和创建紧急修复盘	188	6.1.3 可恢复GPF错误的类型	242
5.3 原始安装	190	6.1.4 安全模式	242
5.3.1 使用FAT-16或NTFS	190	6.1.5 Oracle错误的特征	243
5.3.2 禁止自动启动程序运行	193	6.2 干净的PC机	243
5.3.3 禁止自动启动的服务	194	6.2.1 第1步: 验证文件系统和硬盘的 完整性	244
5.3.4 禁止协议使用和重新绑定排队	196	6.2.2 第2步: 确保简洁的引导	244
5.3.5 整理磁盘碎片	197	6.2.3 第3步: 删除临时文件	245
5.3.6 优化页交换文件	199	6.2.4 第4步: 检查设备间的冲突	246
5.4 优化要考虑的问题	202	6.2.5 第5步: 安装Windows操作系统的 Service Pack	247
5.4.1 禁止BIOS映射	202	6.2.6 第6步: 检查DLL文件的副本	249
5.4.2 检查服务程序包	202	6.2.7 第7步: 检查网络	252
5.4.3 最大化应用性能	203	6.2.8 第8步: 说明虚拟存储器	253
5.4.4 加速引导过程	204	6.2.9 第9步: 安装扩展内存管理器	254
5.4.5 注册表配置编辑器Regedt32和 Regedit	207	6.2.10 第10步: 检查显示设备的配置	255
5.4.6 IDE接口的限制	208	6.2.11 第11步: 对应用进行升级	258
5.5 性能监视工具	209	6.2.12 第12步: 排除ROM寻址空间	258
5.5.1 标准性能监视工具	209	6.3 引发GPF的其他可能原因	258
5.5.2 资源工具箱	220	6.3.1 存储器	258
5.5.3 第三方工具	224	6.3.2 BIOS设置	259
5.6 进行性能调查	227	6.3.3 系统资源	259
5.6.1 使用性能监视器	227	6.4 记录GPF	260
5.6.2 调查网络	230	6.4.1 工具Tweak UI	260
5.7 对Windows 系统的一般提示	231	6.4.2 工具Dr. Watson	263
5.7.1 最佳化Oracle应用的性能	231	6.5 自定义窗体的有关问题	264
5.7.2 工具PowerToys和KernelToys	232	6.5.1 使用构件	264
5.7.3 禁止屏幕保护器	234	6.5.2 设计窗体	265
5.7.4 最佳化Windows NT的低速TCP/IP WAN 连接	234	6.5.3 了解虚拟存储器的使用情况	265
		6.5.4 定制窗体错误	266

6.6 GPF错误调查清单	267	8.3.8 共享缓冲池综述	317
6.7 本章小结	268	8.4 数据存储和布局	319
第7章 优化中间件	272	8.4.1 数据存储	320
7.1 互联计算体系结构	272	8.4.2 数据存储的布局	321
7.1.1 连接进程	272	8.5 理解并配置磁盘I/O	323
7.1.2 规划性能	275	8.5.1 缓冲高速缓存	323
7.2 优化操作系统和硬件	279	8.5.2 对缓冲高速缓存的访问	324
7.2.1 CPU处理器	279	8.5.3 缓冲高速缓存竞争的类型	326
7.2.2 物理存储器和页面交换空间	279	8.6 了解并配置分类	327
7.2.3 TCP/IP协议和操作系统参数	280	8.6.1 Oracle执行分类的方式	327
7.3 优化应用组件	280	8.6.2 Oracle执行分类的时机	328
7.3.1 HTTP服务器	281	8.6.3 为分类分配临时表空间	328
7.3.2 Oracle Web应用服务器版本3.0	281	8.7 并行执行和并行服务器	329
7.3.3 Oracle Apache和WebDB监听程序	282	8.7.1 并行执行	329
7.3.4 优化窗体服务器	287	8.7.2 并行服务器	330
7.3.5 把进程映射到用户	289	8.8 解决数据库性能问题	330
7.3.6 浏览器JInitiator	290	8.8.1 开始调查	331
7.4 故障排除	293	8.8.2 动态性能视图	331
7.4.1 窗体运行时诊断工具	293	8.8.3 验证UtilBstat/UtilEstat报告	332
7.4.2 调试FRM-99999错误	296	8.8.4 分析UtilEstat报告的内容	334
7.4.3 检查窗体、JInitiator和JAR文件	298	8.9 实时性能的测量	340
7.4.4 调查清单	302	8.9.1 共享缓冲池的重访问	340
7.5 本章小结	302	8.9.2 分析资源等待	341
第8章 优化数据库	304	8.9.3 分析SQL区域	344
8.1 数据库基本概念介绍	304	8.9.4 分析游标	345
8.1.1 系统全局区	305	8.9.5 分析锁存器失误	345
8.1.2 执行数据库命令	307	8.9.6 确认运行时间长的SQL语句	346
8.2 数据库统计信息入门	307	8.10 本章小结	346
8.3 了解并配置共享缓冲池	310	第9章 优化服务器	348
8.3.1 共享缓冲池中的SQL和PL/SQL对象	310	9.1 通用服务器部件	348
8.3.2 共享缓冲池的管理方法	311	9.2 调查并优化服务器性能	350
8.3.3 SQL语句的载入和运行	312	9.2.1 监视存储器状态	351
8.3.4 PL/SQL语句的载入和运行	313	9.2.2 磁盘子系统配置和性能	355
8.3.5 控制共享缓冲池中的SQL和PL/SQL 对象	314	9.2.3 CPU处理器的利用率	359
8.3.6 包保持策略	314	9.2.4 监视网络的工作情况	362
8.3.7 识别并解决共享缓冲池竞争问题	315	9.2.5 优化操作系统的核心部分	363
		9.2.6 监视低级系统调用	364

9.2.7 工具一览	365
9.3 性能调查	366
9.3.1 升级服务器	367
9.3.2 主动地进行性能管理	368
9.4 本章小结	369
第10章 优化网络	370
10.1 网络技术简介	370
10.1.1 OSI模型和Oracle应用	370
10.1.2 OSI模型的底层	372
10.1.3 OSI模型的高层	383
10.2 SQL *Net/Net8协议	384
10.2.1 SQL *Net/Net8协议和应用	384
10.2.2 跟踪SQL *Net	387
10.2.3 运行在WAN上的SQL *Net/Net8	388
10.3 Oracle应用中的网络功能	390
10.3.1 连接进程	390
10.3.2 JAR文件和网络流量	391
10.3.3 带宽和延迟	400
10.4 测试Oracle应用	402
10.4.1 可变延迟的GL事务	402
10.4.2 使用测试工具Mercury LoadRunner 和Cloud	404
10.5 测量并模拟网络流量和状态	404
10.5.1 模拟网络运行	405
10.5.2 网络测试工具	408
10.5.3 测量包分段	418
10.5.4 性能基准	419
10.6 服务质量	419
10.6.1 基于应用识别技术的网络	420
10.6.2 虚拟专用网	421
10.7 问题定义	421
10.8 本章小结	422
第11章 优化SQL语句	424
11.1 基于规则和基于成本的优化器	425
11.1.1 基于规则的优化	425
11.1.2 基于成本的优化	427
11.1.3 CBO何时使用参数optimizer_mode =RULE	427

11.1.4 分析CBO的特点	429
11.1.5 分析策略: 分析的时机和方法	431
11.2 理解SQL和索引	433
11.2.1 SQL的处理过程	434
11.2.2 共享和对SQL语句的二次分析	434
11.2.3 非共享SQL语句与共享SQL语句 比较	435
11.2.4 理解索引的功能	436
11.3 调试	440
11.3.1 识别载入频率高的SQL语句	440
11.3.2 开始跟踪	443
11.3.3 SQL跟踪工具	447
11.4 最佳化SQL语句和改善性能	455
11.4.1 测试程序的代码	456
11.4.2 使用提示	457
11.4.3 高水印	461
11.4.4 如何避免全表扫描	463
11.4.5 优化SQL语句	465
11.4.6 优化视图	467
11.5 本章小结	467

第三部分 优化用户和应用子系统

第12章 优化用户	469
12.1 培训程序	469
12.2 提高用户的查询效率	471
12.2.1 内置窗体的限制条件	472
12.2.2 如何查找窗体查询	472
12.2.3 如何使用智能查询	475
12.3 与上下文环境有关的可变字段	477
12.4 本章小结	487
第13章 优化并行处理	488
13.1 并行处理概述	488
13.1.1 并行管理器	489
13.1.2 并程序	491
13.1.3 并行程序库	491
13.2 并行处理的生存期及数据库技术	492
13.2.1 并行处理的生存期	492

13.2.2 并行管理器的生存期	498
13.2.3 并行处理的自动恢复	502
13.2.4 数据库技术回顾	504
13.3 管理并行处理	506
13.3.1 建立定义完整的环境	507
13.3.2 需要监视的对象	511
13.3.3 常用监视工具	511
13.4 处理特殊的性能问题	514
13.4.1 掌握并管理作业提交延迟	515
13.4.2 掌握管理并提高运行效率低的 作业	518
13.4.3 掌握并改善作业终止时出现的 延迟	521
13.4.4 掌握并解决输出延迟	521
13.4.5 对环境的细微调整	522
13.4.6 不断对性能进行评价	524
13.5 本章小结	524
第14章 优化Oracle工作流	526
14.1 Oracle工作流的结构体系	526
14.1.1 工作流开发客户	528
14.1.2 Oracle服务器	529
14.1.3 应用服务器	532
14.1.4 最终用户客户	532
14.2 性能优化	535
14.2.1 信息收集	535
14.2.2 优化性能的建议	539
14.3 排除性能故障	551
14.3.1 工作流引擎中存在的瓶颈	551
14.3.2 丢失的通知	554
14.3.3 挂起的进程	555
14.4 本章小结	556

第四部分 优化案例和回顾

第15章 案例研究	559
15.1 基本信息	559
15.1.1 项目基础	560
15.1.2 Oracle合同和客户合同	560
15.1.3 问题定义	560
15.1.4 系统环境	562
15.1.5 Oracle数据库环境	562
15.1.6 应用环境	563
15.1.7 Web/Forms服务器	564
15.1.8 客户环境	564
15.1.9 网络环境	564
15.1.10 有关附加信息	565
15.1.11 对硬件和环境的要求	565
15.1.12 审查附加信息	566
15.1.13 审阅UtilBstat和UtilEstat报表	566
15.1.14 分析问题	569
15.2 本章小结	571
第16章 本书总结	572
16.1 建立正确的调查方向	572
16.2 人和社会因素对性能的影响	574
16.3 在Oracle的支持下开展工作	574
16.4 与客户协调工作	576
16.5 进入互联网时代	579
16.6 最后的建议	580

第五部分 附 录

附录A 项目文档摘要	581
附录B 记录性能问题	590
附录C 本书光盘内容介绍	594

第一部分 用于调查和编制 应用性能文档的工具

第1章 性能方法论

Oracle Applications（应用）系统通常由若干高度集成并相互关联的模块组成，由于这些模块可以采用不同的方式实现，所以使得Oracle应用系统有时变得相当复杂。通常，构成系统的各种模块都是动态的，用这些模块组成的系统也都各不相同。尽管我们可以在系统设计的初期尽量多地考虑到将来可能发生的问题，但可以肯定地说，在系统生存期的某些阶段，必然要出现某些系统性能的问题。可以想象当我们面对这些具有挑战性的系统性能问题或其他更棘手的问题时，如果我们束手无策，甚至无法确认哪些系统资源对系统性能有影响时，那将要引起一种什么样的结果。本章的目的就是要为读者提供解决这类问题的方法。

尽管我们可以采用各种识别问题的方法、决策支持系统、性能改进技术和各种其他方法来处理Oracle的性能问题，但这些方法不是太抽象就是效果一般。除此之外，这些传统方法的一个共同特点是它们都比较复杂，只有具有专家水平的技术人员才能熟练使用。在某些情况下，有些方法的配置和使用甚至要比解决系统发生的问题的难度还要大。相比之下，本章为读者提供的方法可用来有效地识别并定位Oracle应用系统发生的各类复杂问题。需要说明的是，本章提供的方法不仅适用于Oracle应用系统，而且也可用于诊断其他Oracle数据库系统存在的问题。

提示 本书第4章结尾处的表格向读者提供了一个使用本章方法的案例。

本章所介绍的方法为，把系统作为一个整体来讨论其所有相互作用的组件。虽然由Kepner Tregoe和马里兰大学提出的方法也使用了与本书类似的技术，但其更适用于简单的情况。本书提供的整体方法在继承了其他方法的优点的同时采用了解决问题的五阶段通用规则，即发生了什么问题（What），问题发生的地点（Where），问题发生的时间（When），问题涉及的范围（Extent）和优先级（Priority）。至于读者在实际工作中应采用哪种方法，可自行选择。如果读者使用其他方法或对其他规则比较熟悉的话，你会发现本章对各个阶段的建立与合并是很有帮助的。

本章讲解的方法都是在解决大量系统常见问题的实践经验基础上总结而来的。实践已经证明使用本书提供的方法可以快速有效地定位并解决系统性能问题。我们已经在确保本方法可用来解决复杂的Oracle应用问题的同时，已在大量实验的基础上对这些方法进行了优化。这些经过优化的方法不仅为问题的定位提供了统一的过程，而且还可以发现由系统其他部分的性能问题而引起的副作用和潜在的不良后果。

1.1 性能方法论的引入

通过断开系统资源的方法很少能解决性能问题。这种做法将可能导致单位内部不同部门间

的意见分歧。在这种情况下，系统管理员只能在其中起调解作用。如果某个现场专家认为他所负责的部分没有问题的话，我们就可以使用本章的方法来确认他的结论是否正确。

提示 要养成根据事实做结论的习惯。

解决系统性能问题时通常有两种方法可以采用。一种方法是请精通Oracle系统优化工具的专家来识别并解决问题，另一种方法是通过断开出现问题的硬件。显然，上述两种方法都不是解决问题的根本之道，它们都只是找到了问题现象而不是引发问题的原因。

虽然读者可以借助于多种方法来处理系统性能问题，但本章所描述的方法的价值所在就是读者可以不再需要请专家来定位问题的位置，而由自己来快速有效地缩小问题的范围，进而尽可能快地找到与问题相关的资源。这种方法不仅可以为读者节省大量用在进行系统局部优化所用的时间，而且还可以有效避免因处理系统问题而引起的额外开销。

注意 尽管优化是专门的技术，但它也涉及到许多常识性知识。

1.1.1 读者必须是专家吗

如果读者对每一门计算机技术都精通的话，那是再好也不过了。但这不太现实。通常，我们在遇到问题时都要请专家指导。很显然，请专家帮助有助于快速解决问题。但如果我们具有问题所在领域的专业知识的话，我们就可以自己来解决问题。

借助于本章提供的方法，即使不能立即解决问题，读者也可以快速地确认解决问题所需的专业知识领域和技能范围，进而可确定解决问题的思路。例如，即使没有网络专业知识，读者也可以确定某个复杂网络中存在故障的部件。通过先得到已优化电脑进行某项操作所需的时间，接着把在网络的不同位置接入到电脑并计算同一操作使用的时间。最后通过比较理想时间与实际使用时间之间的差距就可以确定存在网络故障的部件。在处理这类问题时，首先要考虑连接智能客户到服务器的交换机，或者网间计算的中间件部分，这是因为上述部件占据了大部分的处理时间。

1.1.2 了解用户需求

很多Oracle数据库应用系统的用户常把智能客户或网间计算与老式的基于字符处理的大型机系统相比较。这两类设备除了在系统结构方面不同之外，它们之间的最大差别是智能客户使用的图形窗口界面完全改变了传统的操作方式。在某些情况下，从大型机到Windows操作系统的转换已使得某些不熟悉图形用户界面的用户感到很不习惯并企图以各种借口来远离这种智能客户系统。我们将在下面经常提到这个问题。

性能的重要性是不言而喻的，但在考虑性能问题的同时也必须要注意到它所涉及的其他问题。我们可以假设系统在已实现正确配置的情况下，新用户也许会执行某些事先没有考虑成熟的查询命令，从而影响系统的性能。在这种情况下，就不能简单地认为系统计划不周并因此对服务器进行升级，正确的解决方案是要对这些新用户进行培训，从而避免问题的发生。

1.1.3 性能方法论使用的工具

在开始着手解决问题之前，最好利用一点时间来介绍有关系统和系统常见故障的情况。这

是因为如果没有足够的系统信息支持的话，也许会在调查系统故障的过程中走许多弯路。下面要介绍两个用来帮助收集信息和处理故障的工具：

- 要总结相关的系统信息，可以使用下一章中及本书附录A中介绍的项目文档摘要(PDS)。该文档摘要是Oracle 开发部门用来处理复杂性能问题的首选工具。使用该文档结构可以快速构造系统的性能项目文档，完成一份项目文档一般只需要几个小时。如果读者能够利用一点时间来阅读这些文档的具体内容，这将有助于理解我们在本章后面将介绍的优化概念。
- 为了确定性能问题的性质，可以使用本章中介绍的流程图来作为解决问题的框架。由于确定问题的性质是至关重要的，所以这种可以快速确定问题所在范围的手段就显得更有价值。本书假设读者已经对流程图有所了解。

提示 对系统的全面了解将有助于保持解决问题的正确方向。

1.2 掌握系统技术

在开始介绍本方法并将性能问题进行分类之前，我们最好先了解系统的主要部件及其采用的技术，这将有助于我们对性能问题进行分类。例如，如果我们想要增加一个索引，我们就要意识到该索引的引入将导致插入和更新操作的效率下降。这种考虑问题的方法也可以用于对系统性能进行优化的领域。也就是说，当我们考虑可能的解决方案时，我们还要注意到这些方案中将会对系统其他部分产生影响的因素，这些因素中有些是积极因素，有些是消极因素。换句话说，也就是我们不希望以引起新的问题为代价来解决现存的问题。

1.2.1 三种类型的应用系统

如图1-1所示，Oracle应用具有三种不同类型的数据处理过程。这三种应用类型都有其自己特殊的处理要求：

- **联机事务处理（OLTP）** 这种主要用于联机事务处理的系统是面向字符信息的具有大吞吐量和快速插入、更新操作的系统。这种系统中使用的数据库的容量不断增长并具有可观的并发用户。
- **批量数据处理** 大多数批量数据处理模块传送的是由OLTP用户输入的大量数据。通用转账处理就是这种批量数据处理的典型案例。
- **报表处理** 报表通常要对大量的数据进行分类处理，这种操作对系统的要求与批量数据处理的情况相似。编制不周的报表处理程序可能会占用系统可观的内存和大量的中央处理器处理时间。通用转账语句发生器的编制就是一个非常典型的例子，这种程序的代码可能写得非常好，但也有可能非常糟糕。Data Query和Discoverer可能会给系统增加额外的负担。

批量数据处理和报表处理可能会在整个系统的范围内引发性能问题。在理想的情况下，在不延误使用的前提下，这类处理一般应安排在系统的正常工作时间之外或在系统对外停机期间执行。

1.2.2 自己动手发现问题

通过亲自动手调查问题，我们可以发现以往认为所需信息付出代价是物有所值的观点是

不正确的。有这样一个案例，一个组织曾经以可观的费用请一家大公司来为其评估公司使用的网络系统。虽然该组织最后拿到的评估报告非常详尽地列出了各种技术细节，但不幸的是，该报告却没有提及上述三种系统可能发生的瓶颈问题。

经验告诉我们不要轻易放弃那些看起来不重要的信息。除了从专业技术人员那里获取信息外，最终用户也是我们获取有价值信息的一个来源渠道。事实证明最终用户往往可以提供那些对解决问题和加速制定解决方案的非常关键的信息。

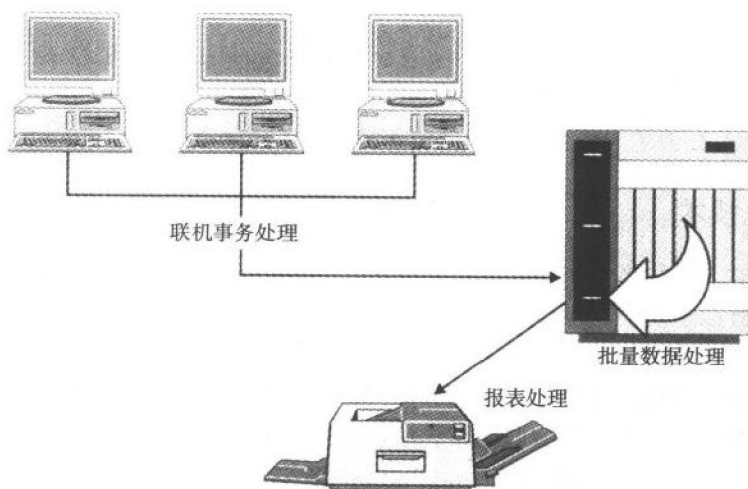


图1-1 三种类型的应用系统示意

1.3 整体化方法

在开始讨论确定问题的五个步骤之前 (What, Where, When, Extent, Priority), 首先来在对整个问题进行调查的环境下考虑这些步骤。在图1-2所示的流程图中我们把调查方法分解为4个阶段：

- **信息收集阶段** 该步骤是调查问题的初级阶段。如果我们能够全面收集并掌握系统的前期性能信息，这将为调查当前问题提供有背景意义的信息，也许这些信息可以成为解决当前的问题的关键线索。
 - **问题调查和优先级确定阶段** 我们在本阶段将对问题进行具体讨论并制定要实现的目标。这是因为如果没有一个合适的目标，我们就可能会漫无边际地进行调查。在实施本阶段的最后几步时，我们要十分谨慎地进行调查，要避免因匆忙下结论而使注意力偏离正确方向。
 - **调查和验证阶段** 该阶段将以实践的方式对所发现的问题进行调查。我们应把注意力放在发现效率最低的部件或处理单元上。这时要在全局的环境下观察问题，以决定是否用另外一种应用或批处理实现该功能。这种考虑方法可能会使我们改变或重新定义性能问题。上述这些工作都应在开始实施系统优化前完成。
 - **优化阶段** 在本阶段，我们的主要工作是对最慢的组件、单元或在前一阶段中发现的其他处理实施优化。我们可能要创建一些测试实例或对系统进行调整并观察这些调整的效果。
- 本章所使用的整体化方法具有简单易行的特点。尽管该方法是针对Oracle应用系统制定的，