

AUTHORIZED ORACLE PRESS™—EXCLUSIVELY FROM OSBORNE

Oracle

性能优化技术内幕

Oracle Performance Tuning 101

Gaja Krishna Vaidyanatha

(美) Kirtikumar Deshpande 著 钟 鸣 石永平 等译

John Kostelac

面对许多书鼓吹的所谓的性能管理“真理”，
这本书无疑是当头棒喝。

——Cary Millsap
HotSOS LLC创始人及总经理



OFFICIAL • AUTHORIZED

Oracle Press

ONLY FROM OSBORNE



机械工业出版社
China Machine Press



OSBORNE

TP311.138

60

Oracle技术系列丛书

Oracle 性能优化技术内幕

Gaja Krishna Vaidyanatha

(美) Kirtikumar Deshpande 著

John Kostelac

钟 鸣 石永平 等译

刘晓霞 审校



机械工业出版社
China Machine Press

本书讲述Oracle性能优化技术。书中对给出的每一种优化方法和措施都从原理、具体步骤、所利用的工具等几个方面进行了详细的阐述，很容易学习和掌握。作者实践经验丰富，具有较高的理论水平，对Oracle的体系结构和内部运行机制极为熟悉，对Oracle性能优化问题有许多独到精辟的见解。书中就多年以来关于性能优化人们一直认为是正确的某些错误观点和方法提出了批判，令人信服地证明了这些观点和方法的荒诞，提出了较为正确的观点和方法。

本书可供具有一定基础的Oracle数据库管理人员和应用编程人员学习，也可以作为对Oracle性能优化进行深入研究的重要参考文献。

Gaja Krishna Vaidyanatha , Kirtikumar Deshpande and John Kostelac: Oracle Performance Tuning 101 (ISBN: 0-07-213145-4) .

Copyright © 2001 by The McGraw-Hill Companies, Inc.

Original language published by The McGraw-Hill Companies, Inc. All rights reserved. No part of this publication may be reproduced or distributed in any means, or stored in a database or retrieval system, without the prior written permission of the publisher.

Simplified Chinese translation edition jointly published by McGraw-Hill Education (Asia) Co. and China Machine Press.

本书中文简体字版由美国麦格劳-希尔教育出版公司授权机械工业出版社出版，未经出版者预先书面许可，不得以任何方式复制或抄袭本书的任何部分。

本书封面贴有McGraw-Hill公司防伪标签，无标签者不得销售。

版权所有，侵权必究。

本书版权登记号：图字：01-2001-4769

图书在版编目（CIP）数据

Oracle性能优化技术内幕/（美）瓦亚娜萨（Vaidyanatha,G.K.）等著；钟鸣等译. -北京：机械工业出版社，2002.5

（Oracle技术系列丛书）

书名原文：Oracle Performance Tuning 101

ISBN 7-111-10102-2

I. O… II. ①瓦… ②钟… III.关系数据库-数据库管理系统，Oracle-系统性能-最佳化 IV. TP311.138

中国版本图书馆CIP数据核字（2002）第017207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：张鸿斌 盖江南

北京第二外国语学院印刷厂印刷·新华书店北京发行所发行

2002年5月第1版第1次印刷

787mm×1092mm 1/16·17.75印张

印数：0 001-4 000册

定价：35.00元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换

序 言

你的Oracle系统的性能是否“极好”？你的回答很可能是“否”，因为你不怕麻烦地翻开了这本书就说明了这一点。不止你一个人有这样的感受。1989年以来我所遇到的数百个Oracle数据库的拥有者都受到过数星期的Oracle系统性能不佳的困扰，没有一个例外。

甚至在满足了要求的系统性能的场合，我和我的同事通常都会发现应用程序所占用的硬件容量有一半纯属浪费。我相信每个正在工作的Oracle系统所占用的硬件资源实际上有20%是不需要的。出现这种程度的低效的原因是能够胜任Oracle系统性能优化的人员极为缺乏。

这种特殊人员的缺乏大约已经有十多年了。当然，Oracle系统很复杂，需要有才干的、工作努力的人员去管理。这个世界上有的是有才干、工作努力的人在驾驶飞机、疾病防治、搞政治等复杂工作方面取得成功。然而，为什么很难找到能胜任Oracle系统性能优化工作的人员呢？

这种人员的短缺是因为出版的关于Oracle性能方面的合适的资料极少。是的，许多数据库管理员的桌子上都堆放有专门讲述Oracle性能的数千页文献资料。但不幸的是，目前已出版的Oracle性能方面的资料实际上约束了你改善Oracle系统性能的能力。

而这本书则不同，它是第一本描述改进了的优化方法的入门教科书，而不是至今还鼓吹新一代分析人员使用旧的基于命中率的分析方法（这种方法十多年来一直在引导Oracle数据库管理员钻牛角尖）的书籍。这是一本关于Oracle性能主题的姗姗来迟的入门书籍，其内容有别于目前仍然占主导地位的优化技术的错误方法。

我喜欢本书有几个原因。我认为对初学者来说，怎样确定性能瓶颈，怎样启动Oracle进行跟踪以及怎样理解跟踪的结果等是最为重要的。而这本书是第一本对这些最重要的性能管理给予优先考虑的书籍。它向性能分析的初学者提供了Oracle核心的工作情况及其所支持技术的相关信息，从而增添了Oracle图书的价值。本书指出了一些性能管理方面的严重过失，这些过失在别的许多书籍中被认为是很恰当的措施。在本书中，你最终会明白为什么重建数据库以消除碎片纯属浪费时间。你会看到基于行选择性的索引效率估计是不可靠的。你会了解到为什么99%的数据库缓冲区高速缓存命中率一般并不代表你的系统正以最高效率运行。优秀的系统性能管理人员最终会认为本书是一本能够彻底改变那么多撰写得很好的文章中给出的某些无益（有时甚至是错误）的劝告的好书。

在建议撰写本书的阶段中，Gaja Krishna Vaidyanatha建议我把这本书作为我希望通过建立hotsos.com来推进的Oracle系统性能革命的一个篇章。很长时间以来，我们受够了一种“由偶然事件决定性能”的说法。自1999年末以来，Hotsos、我们的合作公司（如丹麦的Miracle A/S等）以及我们在世界各地的几位同事承诺为Oracle性能管理人员建立一种信

息质量和可访问性更高的标准。我认为本书能成为改善Oracle系统性能的一个必经的基础台阶。

—Cary Millsap
hotsos.com的创建人及经理

前言

本书的基础内容来自一篇论述Oracle性能管理的文章，这篇文章第一次发表在“International Oracle Users Group—Americas Live! 2000”会议上，然后又在“Oracle OpenWorld 2000”会议上发表时进行了更新。在文章的发表过程中，我们中的多数人都认为任何论述Oracle性能优化的书籍都应该不超过40页。当然，这有点偏激了。考虑到出版商所要求的页面格式、插图以及各种表格，本书页数增加了一个数量级，即400页（英文版）。但每一章的每一节都由许多人做了不止一次的检查，务必使其准确恰当。对于我们来说，撰写本书时感觉到好像又从头开始学了一遍Oracle，因为我们花时间核对了所编写的有关Oracle 8.1.7版的所有内容。

坦率地说，本书中所有内容，包括幽默小段和小故事都是有目的的。我们认为幽默感是这一类技术书籍所需要的；没有幽默，只包含一些呆板信息，所写出的书籍会令人生厌。这本书在许多方面都是与众不同的，我们希望你能欣赏它。

本书的与众不同之处

本书与其他Oracle优化书籍的不同之处在于采用了验证过的方法来处理性能优化。在这样的处理过程中不提供多少脚本（提供太多脚本只会加重问题的复杂性，而不会得出任何关于问题的原因和解决方案的信息）。我们使用了术语“Oracle性能管理”（Oracle Performance Management），因为我们相信Oracle中的优化工作应该视为一种约定，而不是一种随意的搜寻活动。我们不会根据某种随意的缓存命中率或虚构的、随意的传说做出优化建议。

本书所建议的每项优化工作或者是基于对当前Oracle瓶颈的确定，或者是基于事先积极主动地设计和实现Oracle数据库以防止出现这样的瓶颈。本书中所给出的每项内容都已经在一个或多个成品系统上做了实现，或者在具体的正在开发的数据库上进行了充分的测试。本书中没有任何一个优化建议是基于实验室的实验得出的。我们首先要承认自己是工程人员而不是科学家。我们的经验（加起来超过了70年）来自各种平台上实际的成品数据库。

而且，我们拥有像你这样的DBA可达到的性能优化经验。本书中没有给出任何X\$表的引用。为什么会这样呢？我们并未刻意地使优化简单化，事实上优化本来就应该简单的。我们去掉了优化技术表面的神秘面纱，使其成为常识性的素材。祝你阅读愉快！

怎样阅读本书

没有谁会一口气从头到尾读完一本技术书籍。如果对这本书你这样做，将会给我们留下深刻印象。但是，我们希望你某个时刻会读完本书中所有章节。不管你具有多少经验，都一定会在每一章中发现一些新东西或者读到一些独特的内容。请顺序阅读第1至10章。尽可能对第2章多读几遍，因为它汇总了本书的内容，反映出了本书的精髓。第11章和第12章可以在任何时候阅读。第13章给出了本书每一章的概要。

本书共分六个部分。

第一部分：方法

第一部分说明撰写本书的理由，略述Oracle性能管理工作所应遵循的方法。在阅读过第2章后，会获得一些可用来找出现Oracle性能瓶颈的有价值的信息。

第二部分：应用程序优化

第二部分讨论DBA在优化或寻找应用程序中的优化机会时必须了解的问题。第3章描述优化程序、统计数据、提示、SQL连接方法、索引的应用等，用足够的工具武装DBA以处理应用程序中效率低下的SQL问题。第4章介绍tkprof、explain plan和autotrace，在跟踪效率低下的SQL代码方面向DBA及开发人员提供了更多的帮助。为了透彻地理解优化一个应用程序应该采取什么措施，建议仔细阅读这两章。

第三部分：实例与数据库优化

第5章到第8章提供了具体优化Oracle实例各个方面的大量信息。阅读第5章和第6章可掌握Oracle实例的优化，阅读第7章可理解重做日志缓冲区优化并了解某些优化Oracle优化程序本身的方法。第8章有助于优化数据库文件、表、索引等的物理特性。

第四部分：特殊优化

第四部分的两章讨论特殊的优化问题。可以在必要时阅读这两章。第9章讨论并行查询优化问题。第10章介绍数据库中导致争用的原因以及可采取什么措施来避免争用。

第五部分：环境优化

第五部分介绍影响Oracle性能的非Oracle问题。第11章中有关于RAID和I/O配置以及Oracle怎样使用RAID的所有信息。

第12章介绍影响Oracle性能的操作系统优化。该章中讨论了适用于UNIX和Microsoft Windows NT的一般问题，以及适用于Solaris、HP-UX、AIX和Windows NT操作系统的非常特殊的问题。建议在配置系统以驻留一个Oracle数据库之前阅读一下第11和12两章。

第六部分：附录

附录A——词汇表：搜集了本书中使用的所有技术词汇。

附录B——更多的技巧与资源：其中可找到关于优化导出、导入及Oracle的SQL*Loader工具的一些信息。而且还含有一个初始化参数表，为便于理解将它们按功能进行了分组。其中大多数参数在本书各部分进行了介绍。此附录中还有一组包含特定的优化问题的更多信息资源的Web

站点。

参考文献：列出撰写本书时所参考的资料。所给出的Internet URL链接在撰写本书时是最新的。

目 录

序言 前言

第一部分 方 法

第1章 Oracle性能管理介绍	3
1.1 什么是优化	5
1.2 为什么需要优化	5
1.3 由谁来优化	6
1.4 优化到何种程度	6
1.5 何时停止优化	7
1.6 本章小结	8
第2章 优化方法	9
2.1 为什么要重视优化方法	11
2.2 什么是良好的优化方法	12
2.3 Oracle性能优化的基本方法	12
2.3.1 设立合理的性能优化目标	13
2.3.2 测量并记录当前性能	14
2.3.3 确定当前Oracle性能瓶颈	20
2.3.4 把等待事件记入跟踪文件	29
2.3.5 确定当前OS瓶颈	30
2.3.6 优化必需的组件	34
2.3.7 跟踪并实施更改控制过程	35
2.3.8 测量并记录当前性能	35
2.3.9 重复步骤3到步骤7直到满足 优化目标	35
2.4 本章小结	35

第二部分 应用程序优化

第3章 应用程序优化——与DBA有关的问题	39
3.1 Oracle优化程序的故事	41
3.1.1 老伙计：基于规则的优化程序	41
3.1.2 基于规则的优化程序的 不灵活性的影响	42

3.1.3 基于规则的优化程序和C编译器： 一个专家的观点	43
3.1.4 新生儿：基于成本的优化程序	43
3.1.5 基于成本的优化程序的成熟过程	44
3.1.6 幸福老人：基于规则的优化程序	44
3.1.7 回到基于成本的优化程序	44
3.1.8 基于成本的优化程序的崛起	44
3.2 Oracle优化程序的初始参数设置	45
3.3 什么是提示	45
3.4 运行的是哪个优化程序	46
3.5 对象统计数据的计算	47
3.5.1 为什么要计算统计数据	47
3.5.2 怎样计算统计数据	47
3.5.3 需要多少统计信息	47
3.5.4 计算对象统计数据的各种方法	48
3.5.5 间隔多久计算一次统计数据	50
3.5.6 关于计算对象统计数据的问题	51
3.6 最佳检索策略	51
3.6.1 什么是索引	51
3.6.2 何时使用索引	52
3.6.3 怎样建立最佳索引	52
3.6.4 何时需要重建索引	55
3.7 应该使用哪种连接方法，何时使用	56
3.8 怎样不编写SQL	57
3.9 最佳SQL的基础	63
3.10 本章小结	66
第4章 应用程序优化——追踪不良的 SQL语句	67
4.1 SQL语句的优化过程	68
4.2 怎样跟踪SQL	69
4.3 跟踪文件的位置以及如何找到它	70
4.4 在跟踪文件上运行tkprof	72
4.5 解释tkprof中的输出结果	73
4.6 Oracle的行动计划	74

4.6.1 怎样获得Oracle的行动计划	74	6.3.3 回收池	109
4.6.2 如何阅读计划	75	6.3.4 分配对象给一个池子	110
4.7 什么是AUTOTRACE	76	6.3.5 使用cache选项	110
4.8 本章小结	77	6.4 分析数据库缓冲区高速缓存	110
第三部分 实例与数据库优化			
第5章 实例优化——共享池	81	6.4.1 高速缓存命中率	110
5.1 Oracle体系结构	83	6.4.2 数据库缓冲区高速缓存中有什么	112
5.1.1 系统全局区	84	6.4.3 影响数据库缓冲区高速缓存 的等待事件	112
5.1.2 后台进程	86	6.5 解决问题	113
5.1.3 另一个进程：服务器进程	88	6.6 本章小结	114
5.1.4 程序全局区	88	第7章 实例优化——重做日志缓冲区 及杂项优化	117
5.2 分析SQL：用户按ENTER时的工作	89	7.1 配置重做日志缓冲区	118
5.2.1 硬分析与软分析	90	7.1.1 影响重做日志缓冲区的初始 化参数	121
5.2.2 分析或者不分析，这是一个问题	91	7.1.2 影响重做日志缓冲区的等待事件	121
5.3 初始化参数和共享池	91	7.1.3 解决重做日志缓冲区问题	122
5.4 配置各种池子	91	7.2 实例的杂项优化	123
5.4.1 共享池	92	7.2.1 检查点	123
5.4.2 大型池	92	7.2.2 重做日志文件	124
5.4.3 Java池	93	7.2.3 怎样设置重做日志文件的尺寸	124
5.5 优化SPA	93	7.2.4 归档	125
5.6 把对象留在内存中	97	7.2.5 用于实例杂项优化的初始化参数	125
5.7 共享池碎片：主动处理ORA-04031	98	7.3 优化Oracle优化程序	126
5.7.1 导致共享池碎片的原因	98	7.4 本章小结	128
5.7.2 Oracle 7.3及以上版本中的 ORA-04031错误	99	第8章 数据库优化	131
5.8 影响共享池区的等待事件	99	8.1 选取正确的数据库块尺寸	133
5.9 本章小结	100	8.1.1 数据库块尺寸是如何影响性能的	133
第6章 实例优化——数据库缓冲区 高速缓存	103	8.1.2 如何定义最佳Oracle数据库块 的尺寸	133
6.1 什么是五分钟高速缓存规则	105	8.1.3 改变数据库块尺寸：核心问题	135
6.2 数据库缓冲区高速缓存的工作原理	105	8.1.4 小块与大块尺寸的比较：一个 有趣的观点	136
6.2.1 Oracle8i以前的数据库缓冲区 高速缓存管理	106	8.1.5 小结	136
6.2.2 Oracle8i及以上版本中的数据库缓冲 区高速缓存管理	107	8.2 配置块级存储参数	137
6.3 配置缓冲池	108	8.2.1 配置pctused	137
6.3.1 从默认池开始	108	8.2.2 配置pctfree	138
6.3.2 保留池	109	8.2.3 配置initrans	138
		8.2.4 配置maxtrans	139

8.2.5 配置可用列表	139
8.3 设计、配置和优化表空间	140
8.3.1 四个桶表空间的配置方法	140
8.3.2 配置临时表空间	144
8.3.3 全局临时表和临时表空间	144
8.3.4 配置本地管理的表空间	145
8.4 改善性能的数据库分区	146
8.4.1 分区的实际好处	146
8.4.2 数据库分区的关键因素	147
8.5 配置初始化参数	148
8.6 混合型数据库的优化问题	148
8.7 数据仓库数据库的优化问题	150
8.8 本章小结	150

第四部分 特殊优化

第9章 并行查询优化	155
9.1 什么是并行, Oracle如何使用并行	156
9.2 何时使用并行查询	156
9.3 怎样使用并行	157
9.4 能从并行中受益的SQL语句	160
9.5 影响并行性的初始化参数	161
9.6 为并行性设计数据库	164
9.7 并行DML的考虑	165
9.7.1 PDML和回退段配置	165
9.7.2 PDML和实例恢复	165
9.7.3 PDML的限制和问题	166
9.8 并行查询监控	166
9.9 本章小节	168
第10章 争用优化	171
10.1 监控Oracle的争用	172
10.2 回退段	173
10.2.1 什么是多版本读取一致性	173
10.2.2 多版本的读取一致性如何工作	173
10.2.3 定义并揭穿环绕的神话	174
10.2.4 删除回退段争用	176
10.2.5 回退段的用法	178
10.2.6 怎样配置回退段	179
10.2.7 怎样避免“ORA-01555-Snapshot Too Old”错误	181

10.3 主动管理临时段的管理争用	182
10.3.1 理解临时段争用	183
10.3.2 监控临时段表空间的使用情况	184
10.4 死锁	185
10.5 本章小结	186

第五部分 环境优化

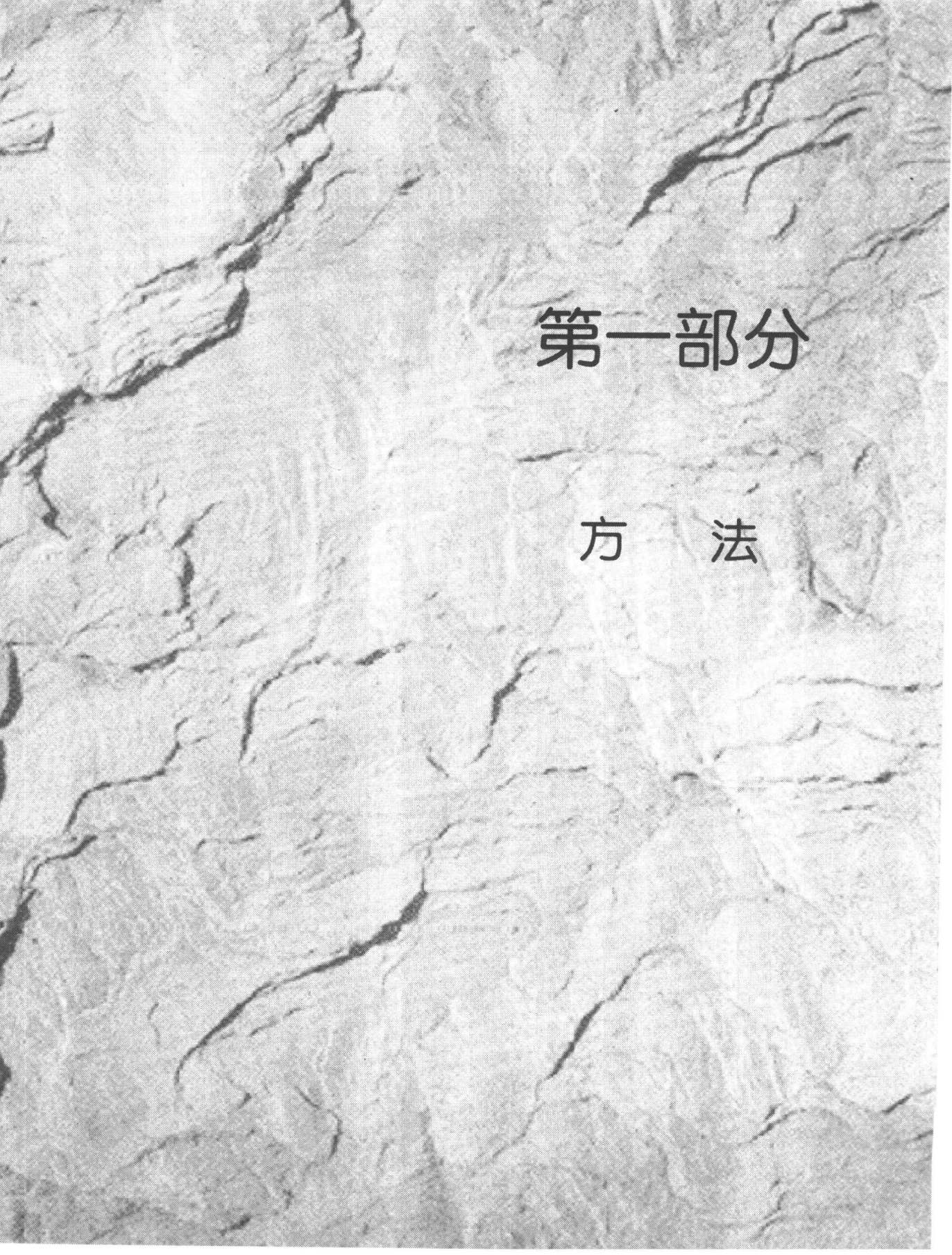
第11章 I/O优化	191
11.1 RAID是什么	192
11.2 RAID不是什么	193
11.3 为什么应该重视RAID	194
11.4 RAID中的三个主要概念	195
11.4.1 什么是条带化	195
11.4.2 什么是镜像	195
11.4.3 什么是等同	196
11.4.4 本节内容提要	196
11.5 RAID的类型	196
11.6 RAID的级别	197
11.6.1 RAID 0	197
11.6.2 RAID 1	197
11.6.3 RAID 0+1	197
11.6.4 RAID 1+0	198
11.6.5 RAID 2	198
11.6.6 RAID 3	198
11.6.7 RAID 4	198
11.6.8 RAID 5	199
11.6.9 RAID 6	199
11.6.10 RAID 7	199
11.6.11 RAID-S	200
11.6.12 Auto RAID	200
11.7 Oracle与RAID	201
11.7.1 RAID 1	201
11.7.2 RAID 0+1和RAID 1+0	201
11.7.3 RAID 3与RAID 5	201
11.7.4 RAID 7	202
11.7.5 Auto RAID	202
11.8 配置磁盘阵列要考虑的基本问题	203
11.9 磁盘条带化要考虑的基本问题	204
11.9.1 建立条带化卷的步骤: 第一部分	205

11.9.2 条带宽度配置	206
11.9.3 建立条带化卷的步骤：第二部分	207
11.10 操作系统配置	207
11.10.1 原始设备与文件系统的严谨讨论	208
11.10.2 异步I/O	208
11.11 数据库的最佳布局	208
11.11.1 并发访问对象的分离	208
11.11.2 把数据与相应的索引分开	209
11.11.3 回退表空间与临时表空间共存	209
11.11.4 分离表空间中的热点对象	209
11.11.5 数据应该怎样条带化	209
11.12 影响I/O性能的初始化参数	210
11.13 RAID和Oracle数据库：核心问题	210
11.14 样例RAID配置	211
11.15 本章小结	216
第12章 操作系统优化	217
12.1 优化OS：一般问题	218
12.1.1 为系统配置足够的内存	219
12.1.2 合理的内存分配方法	220
12.1.3 优化文件系统缓冲区高速缓存	221
12.1.4 优化系统上的交换空间	221
12.1.5 将Oracle SGA锁定在内存中	222
12.2 优化UNIX核心	222
12.3 优化Solaris	224
12.3.1 异步I/O	224
12.3.2 将SGA锁定在内存中	225
12.3.3 优化分页新进程	225
12.4 优化AIX	226
12.4.1 异步I/O	226
12.4.2 在内存中锁定SGA	228
12.4.3 优化分页新进程	228
12.5 优化HP-UX	230
12.5.1 异步I/O	231

12.5.2 在内存中锁定SGA	231
12.5.3 优化文件系统缓冲区高速缓存	231
12.5.4 优化进程管理	232
12.6 优化Windows NT	233
12.6.1 增加Windows NT的可用内存	233
12.6.2 降低前台应用程序的优先级	234
12.6.3 取消不使用的网络协议并重新设置绑定次序	234
12.6.4 配置Windows NT为一个数据库服务器	235
12.6.5 配置“非Windows的打扮”	235
12.6.6 什么是启动开始	235
12.6.7 优化虚拟内存和分页文件	236
12.7 本章小结	236
第13章 全书总结	239
13.1 Oracle性能管理：总结	240
13.2 什么是Oracle性能管理	240
13.3 优化方法	240
13.4 无可替代的应用程序优化	241
13.5 共享池区优化	242
13.6 数据库缓冲区高速缓存优化	242
13.7 重做日志缓冲区及杂项优化	243
13.8 数据库优化	243
13.9 并行查询优化	244
13.10 争用优化	244
13.11 I/O优化	245
13.12 操作系统优化	246

第六部分 附录

附录A 词汇表	251
附录B 更多的技巧与资源	261
附录C 参考文献	269



第一部分

方 法

第1章

Oracle性能管理介绍

误解 如果用更快的CPU升级硬件系统，将会得到更好的性能。

事实 用更快的CPU升级硬件系统试图获得更好性能的方法，在CPU确实不是瓶颈时将会明显地降低性能。这是因为虽然CPU快了但I/O系统并未相应加快，从而导致系统不平衡。例如，如果使CPU的速度加倍，则受到I/O瓶颈牵制的作业有可能要处理两倍的I/O争用。CPU越强处理作业越快，而对I/O资源的要求也就越多。因此在升级CPU前应该仔细考虑一下。

欢迎来到Oracle性能管理的世界。Oracle系统的性能优化技术被认为部分是科学，部分是艺术，部分是巫术。你所拿着的这本书（如果你像大多数DBA那样，你大概会把它放在枕头边）是我们对Oracle的激情以及希望与大家共享信息的一种成果。事实上，我们认为这些信息是如此的重要，以至于宁愿放弃自己对家产的继承权也不会放弃这本书。本书中的工作也是对优化是Oracle的魔法师在暗箱中处理的神密事务这种错误观念的回应。为了破除这些巫术和迷信，使性能管理的可靠原理为大众所用，你应该充分信赖我们。虽然可能你研究的不是航天飞机的多级推进系统，也不是国际空间站的对接模型，不过所有这些原理也可以应用到你的系统之上。毕竟，Oracle性能管理并非高深的火箭科学。

Oracle性能管理是一种利用已经证实过的方法，反复试验、判断和实现优化方案的循序逐进的过程。在从头至尾阅读本书时，会认识到这是一种解决Oracle性能问题的正确方法。我们希望引导你抛弃简单地给Oracle安装更多的共享内存的坏习惯。不要仅因为导师或优化专家说要这样做你就这样做了。

我们希望对在解决故障的方法上带来一些有意义的变革，还希望促使你去分析一些性能问题。最终的目标是消除瓶颈、改善性能，使你有时间多陪陪家人，享受享受生活。业余时间你会问诸如“人生的意义何在？家庭的重要性表现在什么地方？这些人是谁？我真的拥有一个家庭吗？”等问题。如果你想说，“我想我的合作者最好是我的家人。”或者“没有Oracle数据库的生活有何意义？”我想我们就会很合拍了，你大概就会需要这本书了，以及一些专业性的帮助。而且，在把这些原理投入实际的运用前，至少你可以享受到一点新鲜空气、水、阳光和养分。

我们的目的是提供一种方法来优化基于Oracle的系统，而不是仅仅优化某个数据库或某个实例。此外，我们还要揭穿许多常见的有关系统性能和Oracle优化的荒诞的说法。每一章（如这一章）都含有对应于该章内容的“误解”。它们是每个人都“知道”但却是不真实的东西。我们将提供管理整个系统的各种组件的技术，而不仅仅是随意增加Oracle共享池、数据库缓冲高速缓存或重做日志缓冲区的尺寸。

我们将讨论专门针对Oracle主要版本的问题以及与Oracle 7.3及以上版本有关的主题。考察一些重要的平台并提供使用每种平台时所关心的信息。支持你处理优化应用程序的繁琐事务，即使在处理时你没有取得SQL的访问权限也没有关系（如在打包的应用程序中）。除介绍更改初始化参数、简单地建立额外的索引或简单地增加提示以更改查询执行计划外，还介绍了一些别的内容。

本书与其他Oracle优化书籍的一个关键差别是提供了一种经过时间考验的和实地验证过的方

法。这种方法在系统范围内考虑性能管理；它对性能管理采用了一种整体性的方法。虽然新世纪人类可能已经普及了整体方法，但我们可以设想有人刚开始用整体的方法来帮助人类安排自己的身体、灵魂和精神。更多信息请访问www.orapub.com，并参阅由Craig A.Shallahammer撰写的“Total Performance Manager (An introduction to the method)”一文。

通过考察Oracle、操作系统以及应用程序的相互作用，有可能准确地指出瓶颈并得出对性能有很大改善的优化方法。为解决问题，需要缓解瓶颈，目的是彻底地治愈疾病而不只是头痛医头，脚痛医脚。

性能的提高需要量化表示并使用户可察觉得到。如果真正识别和了解了系统的特性、它所受到的约束（最重要的是知道它的瓶颈），则多数可能的性能增益都可以用最小的工作量很快地获得，本书就是专门针对这种思想撰写的。在基于Oracle的系统上管理性能令人激动、沮丧，但同时又是很值得的。利用本书提供的信息，你能够享受到充当一个DBA的令人激动的报酬，而同时又能把挫折减到最少。拥有这份满意的工作，你会让自己的朋友感到羡慕，让自己的仇人感受到了打击。

1.1 什么是优化

优化是有目的地更改系统的一个或多个组件，使其满足一个或多个目标的过程。对Oracle来说，优化是进行有目的的调整组件级以改善性能，即增加吞吐量，减少响应时间。

简单地说，优化表示向Oracle提供适当数目的资源，使用户在完成自己的工作时不用等待所需的数据。后面的说法似乎是很直接了当，即让我们购买市场上最大最可靠的服务器吧。错，虽然这样做可能会解决一定的问题，但不能阻止问题的出现。头痛医头，脚痛医脚是不行的，需要辨证施治。

有时，提供适当的资源并不仅仅表示将更多的内存或CPU装入服务器。它有可能表示重新设计系统组成部分，或恰当地安排工作和批量作业，使它们以一种一致的方式更好地适应已有资源。在性能管理这个范畴内进行优化时，要考虑到系统的各组成部分，对某个部分的需求应该与其他部分均衡。

优化的途径是无穷无尽的。这表示可能要考虑环境中各个方面的问题。这些问题的例子有，删除数据文件或条带化一个表将会怎样影响I/O控制器的用法。或者如果利用NFS文件系统存储某些输出文件（比方说，来自导出或作业的文件），对网络的通信量有何影响？对网络返回数据给交互式用户有何影响？必须考虑对SQL语句所做的更改，考虑它们对客户机或服务器上处理的数据量会产生什么样的影响。因此，性能管理方面的优化意味着为改善性能而最大化所有系统资源的利用。

1.2 为什么需要优化

为什么要优化？用汽车来举例，某个制造商决定要研制一种真正令人激动的高性能汽车。他把技术要求送给了某个设计人员。这个设计人员设计了一张需要最好的组装材料和最强有力的轻便引擎的蓝图。现在，在对预算的制约和轻便引擎进行权衡后，制造人员不得不使用较重的材料和另外的引擎。销售经理提醒制造人员双座跑车或小轿车很流行，因此又做了别的变动。