

The Art and Science of
Oracle Performance Tuning

Oracle

性能优化科学与艺术

(美) Christopher Lawson 著
谢君英 周丽琴 译



清华大学出版社

Oracle 性能优化科学与艺术

(美) Christopher Lawson 著

谢君英 周丽琴 译

清华大学出版社

北 京

内 容 简 介

本书作者根据长期使用 Oracle 的丰富经验,在本书中总结了识别、分离和修复性能问题的五步模型,即从“医师到魔术师”的方法。内容包括性能优化过程、问题分类、优化的人为因素和商业因素、跟踪问题的根本原因、等待事件统计数字的应用、分析 SQL 语句的瓶颈,以及设计和测试解决方案等。此外,本书配有大量的案例分析,使内容更加生动、更有说服力。

本书适合于初/中级 Oracle DBA 和数据库开发人员阅读,读者无需具备有关 Oracle 性能优化方面的背景知识。

EISBN: 1-904347-01-0

The Art and Science of Oracle Performance Tuning

Christopher Lawson

Copyright©2003 by Curlingstone Publishing Ltd.

Original English language edition published by Curlingstone Publishing Ltd.

All Rights Reserved.

本书中文简体字版由 Curlingstone 出版公司授权清华大学出版社出版。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,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

北京市版权局著作权合同登记号 图字:01-2002-0950

图书在版编目(CIP)数据

Oracle 性能优化科学与艺术/(美)罗森 (Lawson, C.) 著;谢君英,周丽琴 译. —北京:清华大学出版社,2004
书名原名: The Art and Science of Oracle Performance Tuning

ISBN 7-302-08188-3

I. O… II. ①罗… ②谢… ③周… 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(2004)第 014374 号

出 版 者:清华大学出版社 地 址:北京清华大学学研大厦
http://www.tup.com.cn 邮 编:100084
社 总 机:010-62770175 客 户 服 务:010-62776969

组稿编辑:曹 康

文稿编辑:王 黎

封面设计:康 博

版式设计:康 博

印 刷 者:北京季蜂印刷有限公司

装 订 者:北京市密云县京文制本装订厂

发 行 者:新华书店总店北京发行所

开 本:185×260 印张:22 字数:563 千字

版 次:2004 年 4 月第 1 版 2004 年 4 月第 1 次印刷

书 号:ISBN 7-302-08188-3/TP·5916

印 数:1~3000

定 价:45.00 元

本书如存在文字不清、漏印以及缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770175-3103 或(010)62795704

前 言

本书以章为单位，循序渐进地介绍了进行性能优化的方法。建议读者先从头到尾通篇阅读全书。一旦熟悉并适应了“医师到魔术师(Physician-to-Magician)”的流程，后面章节中的技术信息就可以作为有用的参考。

使用本书的条件

读者应该理解关系数据库的基本概念和各种类型的 SQL 语句。尽管您不必成为 SQL 专家，但是应该知道连接的含义是什么。

本书还假设您已经具备 Oracle 的基础知识，例如，知道 init.ora 参数是什么，或者侦听器是什么。还应该具备如何启动和停止 Oracle 数据库的经验。

要成功运行本书中的例子，需具备以下条件：

- 一个可运行的 Oracle 实例：8、8i 或者 9i
- SQL*Plus
- 一个具有 DBA 权限的账户

可从 <http://www.curlingstone.com/> 下载本书完整的源代码。

除了这些基础之外，阅读本书不需要性能优化方面的任何背景知识。

客户支持

我们一贯重视读者的意见，并想知道每位读者对本书的看法，包括读者喜欢和不喜欢的内容，以及读者希望我们进一步完善的地方。您可以通过发送电子邮件(feedback@curlingstone.com)来向我们提供反馈意见。请确保在反馈信息中提到本书的书名。

勘误表

我们已经尽最大努力确保本书中的文本和代码不会出现任何错误，但是错误仍然在所难免。如果您发现本书存在错误，例如拼写错误或不正确的代码片段，请给我们反馈信息，我们将不胜感激。发送勘误表可以节约其他读者学习本书的时间，而且能够帮助我们提供更高质量的信息。只要将有关错误的信息以电子邮件的形式发送到 support@curlingstone.com 即可，您的反馈信息将被检查，如果正确，将被粘贴到本书的勘误页面上，并在本书的后续版本中使用该信息。

要想在我们的站点上找到勘误表，请访问 <http://www.curlingstone.com/>，并通过图书标题列表找到本书。

E-mail 支持

如果您希望直接向详细了解本书的专家咨询本书中的问题，可以发送电子邮件到 support@curlingstone.com。这类电子邮件应包括下面的内容：

- 在“主题”字段中包括本书的书名、ISBN 的最后四位数字和出现问题的页码。
- 在信件正文中包括您的姓名、联系方式和问题。

我们不会向您发送无用邮件，因为我们仅仅需要有用的详细资料，以便可节约您和我们的时间。当您发送一个电子邮件信息时，它将经过下面一系列支持：

- 客户支持——首先，您的信息将被递送到我们的用户支持人员手中，由他们亲自阅读。他们将会记录一些频繁提到的问题，并立即回答有关本书或者 Web 站点的任何常见问题。
- 编辑支持——接着，一些较难的问题将被送到对本书负责的技术编辑手中，他们在程序设计语言或者特定的产品方面有着丰富的经验，能够回答相关主题的具体技术问题。
- 作者支持——最后，如果编辑不能回答您的问题(这种情况很少发生)，他们将询问本书的作者。我们将尽量保护作者免受干扰，以便不影响其写作。然而，我们也非常高兴转寄给他们一些特殊的问题。所有 Curlingstone 公司的作者都为他们的书提供技术支持。作为回应，他们将发送电子邮件给用户和编辑，进而使所有的读者受益。

Curlingstone 公司的支持过程仅仅对那些与我们出版的书目内容直接相关的问题提供支持。

目 录

第 1 章 性能优化的挑战	1
1.1 为什么要进行性能优化	2
1.2 编写本书的原因	3
1.3 本书读者对象	3
1.4 读者的好消息	4
1.5 医师到魔术师的方法	4
1.5.1 可适应性的需要	5
1.5.2 理解而非技巧	6
1.5.3 避免猜测	8
1.5.4 银弹	8
1.5.5 解决错误问题	8
1.5.6 揭露一个大秘密	8
1.5.7 与 Oracle 保持同步更新	10
1.6 艺术还是科学	11
1.7 预优化	13
1.8 小结	14
第 2 章 许多优化策略失败的原因	15
2.1 失败的策略	15
2.2 不完善的性能优化方法	16
2.2.1 银弹方法	16
2.2.2 偷工减料	17
2.2.3 快速培训班	22
2.2.4 偏爱的思想	26
2.2.5 解决错误的问题	27
2.2.6 猜测法	30
2.2.7 避免责备	32
2.2.8 其他障碍	33
2.3 小结	34
第 3 章 Oracle 医师	35
3.1 定义问题	36
3.1.1 确定需求	36

3.1.2	定义需求的艺术	36
3.1.3	切忌孤军奋战	37
3.1.4	Oracle 医生的“就职誓言”	38
3.1.5	无需太多的技术	38
3.2	找到主诉症状	39
3.2.1	“数据库是坏的”	40
3.2.2	仔细倾听	41
3.2.3	适时地问一些问题	45
3.2.4	查看所展示的问题	46
3.2.5	必要时寻求帮助	46
3.3	小结	47
第 4 章	处理人际关系	49
4.1	性能优化问题中的人为因素	50
4.2	DBA 的角色	51
4.2.1	集中解决客户的问题	51
4.2.2	采取一个积极的方法	51
4.2.3	建立相互信任的人际关系	53
4.2.4	准确地报告问题	56
4.2.5	避免傲慢的态度	57
4.2.6	承担责任	58
4.2.7	避免责备	60
4.2.8	询问问题	62
4.2.9	接受批评	63
4.3	小结	63
第 5 章	Oracle 侦探	64
5.1	验证问题的存在	65
5.1.1	问题确实存在吗	65
5.1.2	检查基本信息	66
5.1.3	错误消息	66
5.1.4	在控制环境下重新运行应用程序	67
5.1.5	检查运行日志	68
5.1.6	查询 V\$SQL 视图以找到 SQL 语句	69
5.1.7	未使用绑定变量所带来的问题	72
5.1.8	采用工具监控应用程序	75
5.1.9	监视活动和非活动会话	76
5.2	量化问题	76
5.2.1	需要关注的问题	77

5.2.2 方法	77
5.2.3 生成跟踪文件	77
5.2.4 计时信息	80
5.2.5 许可及文件大小	80
5.2.6 TKPROF	81
5.3 注意所有异常因素	90
5.4 小结	90
第 6 章 收集更多的线索	92
6.1 等什么	93
6.2 等待事件的类型	93
6.3 视图概述	97
6.3.1 V\$EVENT_NAME	97
6.3.2 V\$SYSTEM_EVENT	98
6.3.3 V\$SESSION_EVENT	99
6.3.4 V\$SESSION_WAIT	100
6.4 使用系统事件 10046 进行跟踪	102
6.5 等待事件的使用例子	105
6.6 重要的警告	113
6.6.1 不是数据库引起的延迟	113
6.6.2 无法监测 CPU 时间	113
6.6.3 没有监测到的逻辑读	114
6.6.4 时间精确度	114
6.7 小结	114
第 7 章 Oracle 病理学家	115
7.1 隔离根本原因	116
7.2 找出性能降低的根源	117
7.2.1 找出真正的原因	117
7.2.2 性能预算	118
7.3 简化	120
7.4 分治法	124
7.4.1 分化 SQL 代码	124
7.4.2 简化表连接	126
7.5 活动时间表方法	128
7.6 性能问题的常见原因	129
7.6.1 应用程序设计	129
7.6.2 数据库设计	132
7.6.3 init.ora 参数	133

7.6.4	批处理任务的干扰	133
7.6.5	硬件	134
7.7	寻找根本原因的其他建议	136
7.7.1	简单的计时测试	136
7.7.2	大的结果集	136
7.8	小结	137
第 8 章	分析 SQL 瓶颈	138
8.1	SQL 优化的作用	139
8.1.1	SQL 分析人员的好消息	139
8.1.2	SQL 分析的其他好处	140
8.2	使用新的数据库功能	141
8.2.1	需要重新学习一些内容	142
8.2.2	更改过程中的稳定性	142
8.2.3	工具集中的新工具	143
8.3	从哪儿开始	143
8.3.1	查看统计数字	144
8.3.2	产生新的统计数字	144
8.3.3	确保运行的 SQL 可实现预期的功能	146
8.3.4	查看结果集的大小	146
8.3.5	各种优化过程入口的汇总	147
8.4	Oracle 执行计划	147
8.4.1	获得执行计划	148
8.4.2	准备使用 EXPLAIN PLAN 功能	150
8.4.3	运行 EXPLAN PLAN 语句	150
8.4.4	执行计划的内容	151
8.4.5	Oracle 提供的其他脚本	152
8.4.6	使用 SQL * Plus AUTOTRACE 获得执行计划	153
8.4.7	解释执行计划	155
8.5	视图和过滤	159
8.6	子查询	160
8.7	SQL 优化的其他技巧	160
8.7.1	对象浏览工具	160
8.7.2	索引浏览工具	161
8.7.3	注意全表扫描	161
8.7.4	不要忽略逻辑读	162
8.7.5	指导开发人员	163
8.7.6	考虑缓存效果	164

8.8	计划表更为高级的功能	164
8.9	小结	167
第 9 章	分析 SQL 连接	168
9.1	什么是连接	169
9.1.1	等值连接和 θ 连接	169
9.1.2	内连接	170
9.1.3	外连接	170
9.2	连接技巧	171
9.2.1	嵌套循环	172
9.2.2	排序合并	174
9.2.3	哈希连接	176
9.2.4	群集连接	178
9.3	连接优化	179
9.3.1	选择合适的连接	179
9.3.2	SQL 提示	180
9.4	分析连接的图形化方法	182
9.4.1	显示表和连接条件	182
9.4.2	列出过滤统计数字	183
9.4.3	开始连接	184
9.4.4	继续连接	186
9.4.5	多个条件	188
9.4.6	处理到大表的连接	188
9.5	影响连接的 <code>init.ora</code> 参数	189
9.6	其他连接主题	193
9.6.1	连接远程表	193
9.6.2	笛卡尔积	194
9.6.3	数据仓库的特殊连接	195
9.7	小结	201
第 10 章	病理学家的工具箱：其他策略	202
10.1	分析缓慢的数据库	202
10.1.1	是否每个人都觉得性能很慢	203
10.1.2	是否能够将问题归结于一部分原因	203
10.2	找出最差的症状	203
10.2.1	监视活动会话	204
10.2.2	确保没有阻塞会话	205
10.2.3	彻底检查网络	207
10.2.4	查看资源消耗大户	208

10.2.5	磁盘等待时间	210
10.2.6	查看参数极值的设置	210
10.2.7	留心分布式查询	211
10.3	操作系统工具	212
10.3.1	top	212
10.3.2	uptime	213
10.3.3	sar	213
10.3.4	sar 选项	214
10.3.5	使用 sar 查找历史记录	215
10.3.6	iostat	215
10.3.7	vmstat	217
10.3.8	Windows 操作系统中的工具	218
10.4	Oracle Statspack	222
10.4.1	安装 Statspack	222
10.4.2	使用 PERFSTAT 快照	224
10.4.3	运行 Statspack 报告	225
10.4.4	报告示例	226
10.4.5	基础信息	227
10.4.6	缓存大小	227
10.4.7	加载简档	227
10.4.8	实例效率	228
10.4.9	前五个等待事件	229
10.4.10	与前台进程相关的所有等待事件	229
10.4.11	与后台进程相关的所有等待事件	229
10.4.12	实例统计	230
10.4.13	表空间统计数字	230
10.4.14	文件 I/O 统计数字	231
10.4.15	缓冲池统计数字	232
10.4.16	实例恢复统计数字	232
10.4.17	PGA 聚集内存统计数字	233
10.4.18	PGA 内存统计数字	233
10.4.19	回滚段统计数字	233
10.4.20	回滚段存储	234
10.4.21	撤销段摘要	235
10.4.22	锁活动	235
10.4.23	SGA 内存摘要	236
10.4.24	init.ora 参数	236
10.4.25	删除过时的快照数据	237

10.4.26 使用 Statspack 的建议	239
10.5 其他技巧	240
10.5.1 Oracle 9i 对 Statspack 的改进	241
10.5.2 后台揭密：快照的实际工作	242
10.6 小结	242
第 11 章 Oracle 艺术家	243
11.1 绘制解决方案	244
11.1.1 常规解决方案	245
11.1.2 更为复杂问题的解决方案：特殊的性能策略	248
11.2 Oracle Bug	256
11.2.1 难道处处都有 Bug	257
11.2.2 找出大 Bug：报警	257
11.2.3 TAR 的麻烦世界	258
11.2.4 其他排除 Bug 的方法	259
11.3 新的优化选项	260
11.3.1 Database Resource Manager	260
11.3.2 物化视图	269
11.3.3 查询改写	270
11.4 “不可能”的问题	271
11.4.1 与其他 Oracle 专业人员联系	271
11.4.2 咨询系统管理员	272
11.4.3 Metalink	272
11.4.4 Oracle 论坛	273
11.4.5 白皮书	273
11.4.6 有用的 Web 网站	273
11.5 小结	274
第 12 章 艺术家的调色板	275
12.1 SQL 提示	276
12.1.1 何时使用 SQL 提示	276
12.1.2 如何使用提示	277
12.1.3 优化器模式提示	277
12.1.4 表和索引提示	279
12.1.5 连接提示	280
12.1.6 特殊的 SQL 提示	281
12.1.7 SQL 提示疑难解分析	283
12.1.8 语法注意事项	283
12.2 PGA 自动内存管理	283

12.2.1	设置 PGA 目标	284
12.2.2	PGA 目标的优势	284
12.2.3	使用新的 PGA 参数	285
12.2.4	估算 PGA 需求	285
12.2.5	特殊的 PGA 视图	286
12.2.6	特殊的 PGA 术语	286
12.2.7	监视 PGA 内存优化	287
12.2.8	计算新的 PGA 内存目标	288
12.3	柱状图	290
12.3.1	柱状图需求	291
12.3.2	柱状图的样式	292
12.3.3	Oracle 风格的柱状图	292
12.3.4	生成柱状图的传统方法	293
12.3.5	生成柱状图的新方法	294
12.3.6	查看柱状图	295
12.3.7	柱状图的局限性	297
12.4	小结	298
第 13 章	Oracle 魔术师	299
13.1	设置舞台	300
13.1.1	测试解决方案	300
13.1.2	考虑对产品系统的影响	301
13.1.3	保存现有系统	303
13.2	挥动魔棒	304
13.2.1	念动咒语！让所有不满意的用户消失	304
13.2.2	确定性能变化	305
13.2.3	检查统计数据	306
13.2.4	观察意外出现的结果	306
13.3	如果犯了错	307
13.3.1	是否是正确的修正	307
13.3.2	系统如何不同	307
13.3.3	捕获相关的 SQL	310
13.4	做好收尾工作	310
13.4.1	将解决方案记录入档	310
13.4.2	性能改进日志	311
13.4.3	结束与客户的交流	312
13.5	下一步操作	312
13.5.1	公布所学的内容	313

13.5.2 创建性能公文夹	313
13.6 小结	314
第 14 章 案例分析	316
14.1 为什么要花时间进行案例分析	316
14.2 使用案例分析	317
14.3 小结	329
附录 A 有用的 V\$视图定义	330

第1章 性能优化的挑战

本章主要包括：

- 为什么要进行性能优化
- 编写本书的原因
- 本书读者对象
- 读者的好消息
- “医师到魔术师”的方法
- 艺术还是科学
- 预优化

数据库管理员(或者说 DBA)执行的任务非常广泛——一些任务让人兴奋，而其他更多的则是日常事务。在大多数公司里，DBA 并没有选择执行哪些任务的自由。一般来说，所有的 DBA 都必须完成各种任务，其中一些是令人愉快的，另一些则是乏味的。

从消极的一面讲，没有太多的 DBA 会对建立备份、监视磁盘空间，或者查看前一晚上的导出是否运行良好等任务而真正兴奋。对于大多数 DBA 而言，这些任务不成其为挑战，一般也不需要太多的创造性。然而这些任务对于确保一个公司所依赖的重要系统的完整性却是非常重要的。

DBA 的工作还有另一方面，但是其中许多是额外的要求。从积极的一面讲，性能优化为 DBA 提供了一个全新的舞台，他们可以在这里展示自己的魅力。他们的工作不再仅仅是日常事务或者是乏味的操作，性能优化允许分析人员发挥自己的创造性和智慧，并解决影响公司成功运作的一些障碍。

本章将给出性能优化的一个全景图。我们探讨了这样一个主张，即性能优化与其说是一门科学，倒不如是艺术与科学的结合。这意味着 DBA 必须不仅仅是一个可以解决复杂问题的极好的数学家。相反，性能专家必须具有灵活处理过程的能力，使他可以在必要时改变行动方针。

本书将为性能优化提供一个可逐步适应的方法，我形象地称之为“医师到魔术师”方法。利用该方法，DBA 在性能优化过程的各个阶段将扮演不同的角色。每个阶段都有各自独特的任务，适合于那一阶段完成。

本书的实际场景将展示 Oracle 性能优化的挑战和激动人心之处。本章以及全书会尽可能地使用实际的**案例分析**。其原因如下：

- 实际示例通常比我们所能想象到的更晦涩、复杂。
- 以往的例子可以用于阐述一个合理的方法(或者要避免的失误)。
- 生动的案例比纯粹理论上的定理要容易记住一些，并且通常要有趣得多。
- 实际案例的解决方案可以为读者提供类似问题的答案。

本章除了给出各种案例分析之外，还介绍了我们的概要性的性能优化提示。这些提示是笔者认为在理解性能优化过程中比较重要的一些观察和建议。有些提示一开始似乎就很明显，但是许多这样的常识性技巧并没有得到具体的应用。

在介绍“医师到魔术师”方法的同时，我们还讨论了专家会遇到的有关性能优化方面的一些障碍。这些障碍包括“银弹”倾向，以及“猜测方法”。第2章将进一步介绍这些烦人的障碍。

最后，我们以“预防用药”的讨论结束本章。理想情况下，DBA、设计人员和开发人员应尽量避免现在经常出现的许多性能问题。

1.1 为什么要进行性能优化

与一些日常性 DBA 任务相比，性能优化是数据库工作中比较吸引人的任务。但是，多数人并没有认识到性能优化工作的重要价值。性能优化是 DBA 所面临的最具挑战性的课题之一，它将带来切实而不太明显的回报。有很多工作提供这样的机会，使 DBA 因为修复了一家大型公司的重要系统而节约了宝贵时间，从而成为公司的“英雄”？但这只是一个有经验的性能专家才能完成的工作。

人们对性能优化技术也有着较高的需求。图 1-1 演示了雇主在聘用新 Oracle DBA 所要求的一些技术。注意，性能优化比诸如备份和恢复这些基本技术所提到的频率更高。所有要聘用 Oracle DBA 的雇主中，有 42% 的人提到性能优化是必需的技术。

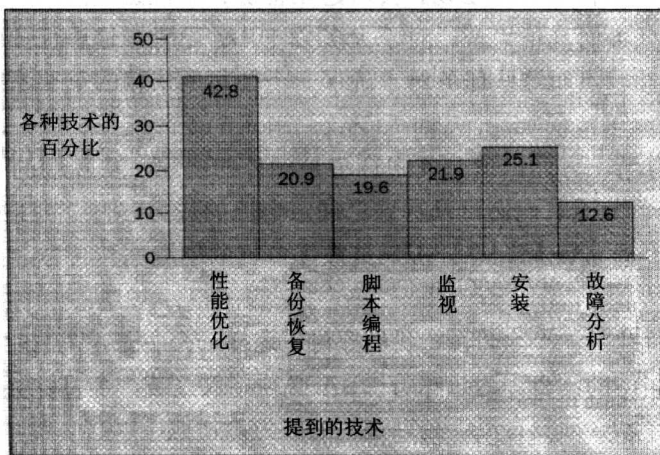


图 1-1

本图来源：DICE Web 站点(<http://www.dice.com/>)，2002 年 3 月 7 日。取样人数：593 个雇主。

DICE(Data Processing Independent Contractor Exchange)是软件专业人员的一个流行 Web 站点。

这些数字可以有几种解释。一种可能是，需要用这些技术来解决一些新发现的。未经优化专家处理的性能问题。也可能是解决一些已经处理过但没有成功的性能问题。不管怎么解释，很明显，人们都很需要能够胜任性能优化工作的 Oracle DBA。

注意，像备份和恢复这类技术的需求稍低一些，但这并不意味着这些技术对公司不太重要。上图只是反映了能胜任优化工作的人员的欠缺。

1.2 编写本书的原因

尽管许多 DBA 和设计人员面临着 Oracle 性能优化的挑战，但是许多人并不真正确定该从何处着手。他们知道必须做一些事情，但是要做哪些事情呢？是否应该首先更改 `init.ora` 参数？或者应该添加更多的数据文件？

如果没有一个良好的性能优化方法的话，哪一种猜测都是一样的，并且很难达到目的。这种猜测方法一开始可能有趣，但是老板很少会对这样得到的结果感到满意。我们都需要一个有效的性能优化的逻辑方法，并得到相应的结果。不管性能分析人员是永久的雇员还是顾问，这都是一个合理的目标。

本书主要面向那些想学习并应用一种系统的、循序渐近的方法来解决困难的性能问题的读者。本书的论点是积极的：即不需要使用猜测、摸索的方法进行性能优化。

当然，当我们谈到良好方法时，并不表示技术内容和经验是不重要的。相反，没有 Oracle 数据库操作方面的良好背景，世界上最好的方法也是无用的。在本书的前面部分我们强调了一个良好的过程，而在后面章节中我们介绍了一些分析方法。这些方法包括如何使用 `V$WAIT`、如何解决表联结、如何使用客户资源组(Consumer Resource Groups)等。

快速了解术语 DBA

全书在谈到解决 Oracle 性能问题的专家时，将使用术语 **DBA** 或数据库管理员。实际上，这人可能根本不是 DBA——性能分析人员可能是开发人员的领导、高级设计人员，或者是程序员。根据不同的组织机构和项目，公司的“Oracle 魔术师”可以是其中任何一种身份。

重要的不是这个人是否叫做 DBA，而是他对数据库基本原理的理解如何，以及他能否很好地应用一致的逻辑过程来解决性能问题。性能优化游戏注重的毕竟是结果。当一个困难的性能问题解决之后，管理层会非常高兴，而不管这个人的头衔是什么。

1.3 本书读者对象

本书主要针对两类读者。第一类包括那些只具有有限的或者中等程度的 Oracle 性能优化经验的 DBA。也许您只担任了很短一段时间的 DBA，或者还在努力学习如何开始解决性能问题，那么本书就是为您量身定制。我们将展示如何开始进行 Oracle 性能优化，以及如何最大化您成功的机会。

除了 DBA 新手之外，本书还将面向那些已经具有一定解决 Oracle 性能问题方面经验的读者。您可能已经分析过一些简单的性能问题。您也许通过添加一两个索引或者重写一些定义不好的 SQL 而成功地解决了一些问题。尽管您已经取得了一些成功，但是觉得自己仍然需要加强 Oracle 优化方面的技术。您希望处理一些更困难的、真正有挑战性的性能问题。如果您的背景和愿望的确如此，那么本书也是为您而写的。

如果您是一位有经验的 DBA，您也可从有关 Oracle 性能优化方面近期一些改进的讨论中获益。例如，我们将探讨一些复杂的技术，比如使用等待事件(wait even)工具来隔离出瓶颈操