

Oracle Database 11g Performance Tuning Recipes
A Problem-Solution Approach

Oracle Database 11g 性能优化攻略

- 问题分析由表及里，调优方案标本兼治
- 示例丰富全面，常见性能问题各个击破
- Oracle数据库管理和开发人员必读

[美] Sam R. Alapati Darl Kuhn Bill Padfield 著
朱浩波 译



人民邮电出版社
POSTS & TELECOM PRESS

013025776

TP311.1380R
313

TURING

图灵程序设计丛书 数据库系列

Oracle Database 11g Performance Tuning Recipes
A Problem-Solution Approach

Oracle Database 11g 性能优化攻略



[美] Sam R. Alapati

Darl Kuhn

Bill Padfield

著

朱浩波

译



北航

C1633658

TP311.1380R
313

人民邮电出版社

北京

图书在版编目 (C I P) 数据

Oracle Database 11g性能优化攻略 / (美) 阿拉帕提 (Alapati, S. R.), (美) 库恩 (Kuhn, D.), (美) 帕德菲尔德 (Padfield, B.) 著; 朱浩波译. — 北京: 人民邮电出版社, 2013. 4

(图灵程序设计丛书)

书名原文: Oracle Database 11g Performance Tuning Recipes: A Problem-Solution Approach
ISBN 978-7-115-30936-5

I. ①0… II. ①阿… ②库… ③帕… ④朱… III. ①关系数据库系统 IV. ①TP311.138

中国版本图书馆CIP数据核字 (2013) 第025084号

内 容 提 要

三位经验丰富的顶级 Oracle DBA 再次联手, 为读者呈现这本 Oracle 数据库性能优化攻略。本书由表及里地深入分析了造成 Oracle 数据库性能缓慢的各种原因, 然后给出标本兼治的性能调优方案。作者将多年的实践经验和个人智慧与读者分享, 帮助读者诊治影响数据库性能的各种疑难杂症。

本书可帮助数据库管理员解决各种 Oracle 数据库性能问题。

图灵程序设计丛书

Oracle Database 11g 性能优化攻略

- ◆ 著 [美] Sam R. Alapati Darl Kuhn Bill Padfield
- 译 朱浩波
- 责任编辑 王军花
- 执行编辑 李 静
- ◆ 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 电子邮件 315@ptpress.com.cn
- 网址 <http://www.ptpress.com.cn>
- 北京天宇星印刷厂印刷
- ◆ 开本: 800×1000 1/16
- 印张: 30.5
- 字数: 721千字 2013年4月第1版
- 印数: 1-3 500册 2013年4月北京第1次印刷

著作权合同登记号 图字: 01-2011-8033号

ISBN 978-7-115-30936-5

定价: 99.00元

读者服务热线: (010)51095186转604 印装质量热线: (010)67129223

反盗版热线: (010)67171154

站在巨人的肩上
Standing on Shoulders of Giants



www.ituring.com.cn

版 权 声 明

Original English language edition, entitled *Oracle Database 11g Performance Tuning Recipes: A Problem-Solution Approach* by Sam R. Alapati, Darl Kuhn and Bill Padfield, published by Apress, 2855 Telegraph Avenue, Suite 600, Berkeley, CA 94705 USA.

Copyright © 2011 by Sam R. Alapati, Darl Kuhn and Bill Padfield. Simplified Chinese-language edition copyright © 2013 by Posts & Telecom Press. All rights reserved.

本书中文简体字版由Apress L.P.授权人民邮电出版社独家出版。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。

献词

献给 Valerie、Nina 和 Nicholas，我爱你们。

——Sam R. Alapati

献给 Heidi、Brandi 和 Lisa。

——Darl Kuhn

献给 Oyuna 和 Evan，感谢你们容忍我每个晚上和周末都坐在电脑前面，而不是陪伴你们。

献给我的家人 Carol、Gerry、Susan、Doug、Scott、Chris、Jaimie、Katie、Jenny、Jeremy 和 Sean。我爱你们！

——Bill Padfield

致 谢

感谢 Apress 杰出的出版团队在本书编写过程中给予我们的帮助。高级组稿编辑 Jonathan Gennick 协助拟定了本书的主题（攻略），还提出了很多深刻的意见和建议，极大地改进了本书表达方式并提高了内容的质量。Jonathan 是少有的编辑，不仅精通技术，而且编辑技能也很精湛。Jonathan，谢谢你在写作过程中的耐心和辛勤工作！在过去几个月中，你睿智的建议和不断的鼓励令我们三个获益匪浅。

还要感谢本书技术编辑，True Internet 公司的高级分析师 Surachart Opun 所做的大量工作。他在撰写博客和其他工作之余拨冗审阅了书稿。Surachart 不仅指出了书中代码和其他地方的几处错误，而且还给出了很多改进各个攻略讲述方式的建议。谢谢你 Surachart，谢谢你为本书的出版所做的细致且充满激情的工作。协调编辑 Anita Castro 在整个过程中给了我们非常好的指导，确保写作能够按计划进行。不管怎么说管理好有三个作者的项目都不是一件轻松的事，但 Anita 显然做到了这一点！Mary Ann Fugate 用娴熟的技能完成了本书的文字编辑工作，我们谢谢她为提高本书质量所做出的贡献。

Sam R. Alapati 的致谢

首先，衷心感谢我的两位合著者 Darl Kuhn 和 Bill Padfield 的极大帮助，和你们合作真的很棒，我每一分钟都很享受！还要感谢我所在的公司——位于新泽西州木桥市（Woodbridge）的 Miro 咨询公司的支持和鼓励。公司首席执行官 Scott Rosenberg 先生不仅是一位好领导，并且以极大的热情在美国各地的客户中推广 Oracle 技术。Miro 公司董事长 Eliot Colon、技术服务副总裁 Wayne Federico，以及副总裁 Bob Kinkade 一直以来对于我在 Miro 的工作给予了极大的支持，我在和他们一起工作的过程中也学到了很多。

感谢我的朋友 Kishore Rachamalla、Praveen Katapally 和 Sreeny Chinta 在我任职于德克萨斯州泰勒市的 ERCOT 公司期间对我的无私帮助，在那里我开始了本书和另外一本书的写作。还要感谢 Sam Nataros 的帮助和支持，我将永远珍惜他给我的心灵之礼——谢谢你，Sam，你的态度每天都激励着我！

当然，我的家人在本书的写作期间牺牲最大。因此我要谢谢 Valerie、Shannon、Nicholas 和 Nina，谢谢他们在过去几年里对我写作本书和另一本书的帮助和支持。最后但同样重要的，我要感谢其他家庭成员——我的母亲 Swarna Kumari、父亲 Appa Rao 和我的兄弟 Hari Hara Prasad、

Siva Sankara Prasad、Aruna、Vanaja、Ashwin、Teja、Aparna 以及 Soumya，感谢他们对我不变的爱和永远的信任。

Darl Kuhn 的致谢

谢谢合著者 Sam Alapati 和 Bill Padfield，同时感谢这些年中传授我性能调优技术的数据库管理员：Dave Jennings、Bob Suehrstedt、Scott Schulze、Pete Mullineaux、Janet Bacon、Sue Wagner、Mohan Koneru、Arup Nanda、Charles Kim、Bernard Lopuz、Barb Sannwald、Tim Gorman、Shawn Heisdorffer、Doug Davis、Sujit Pattanaik、Ken Roberts、Roger Murphy、Mehran Sowdaey、Kevin Bayer、Dan Fink、Guido Handley、Margaret Carson、Nehru Kaja、Tim Colbert、Glenn Balanoff、Bob Mason、Shari Plantz-Masters、Mike Nims、Denise Duncan、Brad Blake、Ravi Narayanaswamy、Abid Malik、Abdul Ebadi、Kevin Hoyt、Trent Sherman、Sandra Montijo、Jim Secor、Maureen Frazzini、Sean Best、Stephan Haisley、Geoff Strebel、Patrick Gates、Krish Hariharan、Buzzy Cheadle、Mark Blair、Gary Dodge、Karen Kappler、Mike Hutchinson、Liz Brill、Ennio Murrone、Mike O'Neill、Beth Loker、Mike Eason、Greg Roberts、Debbie Earman、Tom Wheltle、Ken Toney、Gabor Gyurovsky、Scott Norris、Joey Canlas、Eric Wendelin、Gary Smith、Mark Lutze、Kevin Quinlivan、Dave Bourque、Roy Backstrom、Larry Carpenter、Joe Meeks、Ashish Ray、John Lilly、Dave Wood、Laurie Bourgeois、Steve Buckmelter、Casey Costley、John DiVirgilio、John Goggin、Brett Guy、Simon Ip、Pascal Ledru、Kevin O'Grady、Peter Schow、Todd Sherman、Jeff Shoup、Mike Tanaka、Todd Wichers、Doug Cushing、Will Thornburg、Steve Roughton、Ambereen Pasha、Dinesh Neelay、Kye Bae、Thom Chumley、Jeff Sherard、Dona Smith、Erik Jasiak、Gary Schut、Don Gritzmacher、Aaron Isom、Kristi Jackson、Karolyn Vowles、Amin Jiwani、Paula Still、K. P. Muthe、Joe Pinkerton、Arvin Kuhn、Darin Christensen、Terry Roam、Doug Drake、Marilyn Wenzel、Doc Heppler、Mert Lovell、Carl Beasley、Brian Beasley、Odean Bowler 和 Jim Stark。

Bill Padfield 的致谢

我想感谢两位亲密的合著者 Sam R. Alapati 和 Darl Kuhn，感谢你们接纳我这个写作新手，以及对我的帮助和支持。没有你们的帮助我就没法完成这项任务。

在我的职业生涯中，有这么多曾帮助我的人值得我感谢，因此请相信我感谢每一位帮助和鼓励过我的人。首先感谢 Bob Ranney 让我有机会成为一名数据库管理员。还要感谢一些领导一直以来对我的帮助，他们是 Beth Bowen、Larry Wyzgala、John Zlamal、Linda Scheldrup、Amy Neff 和 Maureen Frazzini。

当然，在我的职业生涯中，还有很多数据库管理员、开发人员、系统管理员以及架构师给了我很大的帮助。首先我要感谢目前工作团队中的数据库管理员们，他们让我每天都充满激情。这些同事曾经给予我很多专业的帮助，并且这么多年一起工作的过程中，我们成为了非常好的朋友。

他们是 Dave Carter、Debbie Fitzgerald、Pankaj Guleria、Pete Sardaczuk、Brad Storm 和 Rebecca Western。

多年来,我还从下面名单中列出的人那里学到了很多,他们一直以来慷慨地花费大量时间帮助我,并且耐心地解答我的问题: Mark Nold、Mick McMahon、Sandra Montijo、Jerry Sanderson、Glen Sanderson、Jose Fernandez、Mike Hammontre、Pat Cain、Dave Steep、Gary Whiting、Ron Fullmer、Becky Enter、John Weber、Avanish Gupta、Scott Bunker、Paul Mayes、Bill Read、Rod Ermish、Rick Barry、Sun Yang、Sue Wagner、Glenn Balanoff、Linda Lee Burau、Deborah Lieou-McCall、Bob Zumpf、Kristi Sargent、Sandy Hass、George Huner、Pad Kail、Curtis Gay、Ross Bartholomay、Carol Rosenow、Scott Richards、Sheryl Gross、Lachelle Shambe、John Piel、Rob Grote、Rex Ellis、Zane Warton、Steve Pearson、Jim Barclay、Jason Hermstad、Shari Plantz-Masters、Denise Duncan、Bob Mason、Brad Blake、Mike Nims、Cathie Wilson、Rob Coates、Shirley Amend、Rob Bushlack、Cindy Patterson、Debbie Chartier、Blair Christensen、Meera Ganesan 和 Kedar Panda。

目 录

第 1 章 优化表性能.....1	攻略 2-13 创建位图连接索引.....59
攻略 1-1 创建具有最优性能的数据库.....1	攻略 2-14 创建索引组织表.....60
攻略 1-2 创建具有最优性能的表空间.....4	攻略 2-15 监控索引使用.....62
攻略 1-3 匹配表类型与业务需求.....6	攻略 2-16 索引创建速度最大化.....63
攻略 1-4 选择有益于性能的表特性.....8	攻略 2-17 回收未使用的索引空间.....64
攻略 1-5 在创建数据表时避免盘区分配 延迟.....9	第 3 章 优化实例内存.....68
攻略 1-6 数据加载速度最大化.....11	攻略 3-1 自动内存管理.....68
攻略 1-7 高效移除表中数据.....13	攻略 3-2 管理多个缓冲池.....71
攻略 1-8 显示自动段顾问建议.....15	攻略 3-3 设定内存最小值.....73
攻略 1-9 手工生成段顾问建议.....18	攻略 3-4 监控内存调整操作.....74
攻略 1-10 自动发送段顾问输出电子邮件.....22	攻略 3-5 优化内存使用.....75
攻略 1-11 重建跨多个数据块的数据行.....23	攻略 3-6 调优 PGA 内存分配.....77
攻略 1-12 释放未使用的表存储空间.....26	攻略 3-7 配置服务器查询缓存.....79
攻略 1-13 压缩数据以进行直接路径加载.....27	攻略 3-8 管理服务器结果缓存.....81
攻略 1-14 为所有 DML 操作压缩数据.....30	攻略 3-9 缓存 SQL 查询结果.....83
攻略 1-15 在列级压缩数据.....31	攻略 3-10 缓存客户端结果集.....86
攻略 1-16 监控表使用率.....33	攻略 3-11 缓存 PL/SQL 函数结果.....88
第 2 章 选择和优化索引.....35	攻略 3-12 配置 Oracle 数据库 智能闪存缓存.....91
攻略 2-1 理解 B 树索引.....36	攻略 3-13 调节重做日志缓冲区.....93
攻略 2-2 选择需要建立索引的列.....41	第 4 章 监控系统性能.....95
攻略 2-3 创建主键索引.....44	攻略 4-1 实现 AWR.....95
攻略 2-4 创建唯一索引.....45	攻略 4-2 修改统计信息时间间隔 和保存期限.....97
攻略 2-5 为外键列创建索引.....48	攻略 4-3 手工生成 AWR 报表.....99
攻略 2-6 何时使用组合索引.....49	攻略 4-4 通过企业管理器生成一份 AWR 报告.....101
攻略 2-7 通过压缩减少索引大小.....51	攻略 4-5 为一条 SQL 语句生成 AWR 报告.....102
攻略 2-8 实现基于函数的索引.....52	攻略 4-6 为数据库创建统计基线.....103
攻略 2-9 在虚拟列上创建索引.....54	攻略 4-7 通过企业管理器管理 AWR 基线.....106
攻略 2-10 平衡索引的 I/O.....56	
攻略 2-11 新增一个索引而不影响 已有的应用.....57	
攻略 2-12 创建支持星型架构的 位图索引.....58	

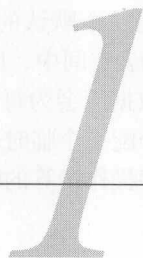
攻略 4-8 管理 AWR 统计信息库	109
攻略 4-9 自动创建 AWR 基线	110
攻略 4-10 快速分析 AWR 输出	112
攻略 4-11 手工获取活动会话信息	113
攻略 4-12 从企业管理器中获取 ASH 信息	117
攻略 4-13 从数据字典中获取 ASH 信息	119
第 5 章 最小化系统资源争夺	123
攻略 5-1 理解响应时间	123
攻略 5-2 确定引起最多等待的 SQL 语句	126
攻略 5-3 分析等待事件	126
攻略 5-4 理解等待事件的分类	128
攻略 5-5 检查会话等待	128
攻略 5-6 按类型检查等待事件	130
攻略 5-7 解决缓冲区忙等待	132
攻略 5-8 解决日志文件同步等待	134
攻略 5-9 被另一个会话读取等待事件的 最小化	135
攻略 5-10 减少直接路径读取等待事件	136
攻略 5-11 恢复写入器等待最小化	137
攻略 5-12 找出谁持有阻塞锁	138
攻略 5-13 确定被阻塞和引起 阻塞的会话	140
攻略 5-14 处理引起阻塞的锁	141
攻略 5-15 确定被锁定的对象	142
攻略 5-16 解决 enq:TM 锁资源争夺	143
攻略 5-17 确定最近被锁住的会话	145
攻略 5-18 分析数据库中最最近的等待 事件	147
攻略 5-19 确定由于锁定所花费的等待 时间	149
攻略 5-20 锁存器争夺的最小化	151
攻略 5-21 通过 Oracle 企业管理器来管 理锁	154
攻略 5-22 通过 Oracle 企业管理器分析 等待	155
第 6 章 分析操作系统性能	157
攻略 6-1 检测磁盘空间问题	159
攻略 6-2 确定系统瓶颈	161
攻略 6-3 确定系统瓶颈 (Solaris)	163
攻略 6-4 确定消耗服务器资源 最多的进程 (top)	164
攻略 6-5 确定 CPU 和内存瓶颈	166
攻略 6-6 确定 I/O 瓶颈	167
攻略 6-7 识别网络密集型进程	170
攻略 6-8 检修数据库网络连接性	171
攻略 6-9 将一个资源密集型进程映射到 一个数据库进程	172
攻略 6-10 终止一个资源密集型进程	175
第 7 章 检修数据库	177
攻略 7-1 确定最优的撤销保留时间	177
攻略 7-2 找出是什么消耗了最多的撤销 空间	181
攻略 7-3 解决 ORA-01555 错误	182
攻略 7-4 监控临时表空间使用率	184
攻略 7-5 确定是谁在使用临时表空间	185
攻略 7-6 解决“无法扩展临时数据段” 错误	186
攻略 7-7 解决打开游标错误	188
攻略 7-8 解决被挂起的数据库问题	190
攻略 7-9 激活自动诊断库命令解释器	194
攻略 7-10 从 ADRCI 中来查看报警 日志	198
攻略 7-11 使用 ADRCI 查看事件	200
攻略 7-12 将事件打包发给 Oracle 技术支持团队	202
攻略 7-13 运行一次数据库健康检查	203
攻略 7-14 创建 SQL 测试用例	205
攻略 7-15 生成一份 AWR 报告	208
攻略 7-16 比较两个阶段的数据库性能	210
攻略 7-17 分析一份 AWR 报告	212
第 8 章 创建高效的 SQL	216
攻略 8-1 获取一张表中的所有数据行	216
攻略 8-2 获取一张表中的部分数据行	218
攻略 8-3 通过相对应的行来连接表	220
攻略 8-4 在没有相对数据行的情况下 连接表	222
攻略 8-5 构造简单的子查询	224
攻略 8-6 构建相关子查询	228
攻略 8-7 比较两个表找出缺失的数据行	230
攻略 8-8 比较两张表找出匹配的数据行	231

攻略 8-9 将相似 SELECT 语句的结果 集合并.....	232	攻略 10-12 在 RAC 系统中追踪并行查询	295
攻略 8-10 查找一定范围内的值	234	攻略 10-13 合并多个追踪文件	296
攻略 8-11 处理空值	237	攻略 10-14 找出正确的会话来进行追踪	297
攻略 8-12 搜索部分列值	240	攻略 10-15 追踪一个 SQL 会话	297
攻略 8-13 重用共享池中的 SQL 语句	243	攻略 10-16 通过进程 ID 来追踪会话	299
攻略 8-14 避免偶然的全表扫描	246	攻略 10-17 追踪多个会话	300
攻略 8-15 创建高效的临时视图	248	攻略 10-18 追踪一个实例或数据库	301
攻略 8-16 避免使用 NOT 子句	250	攻略 10-19 为会话生成事件 10046 追踪	302
攻略 8-17 控制事务大小	252	攻略 10-20 为实例生成事件 10046 追踪	304
第 9 章 SQL 手工调优	255	攻略 10-21 在一个正在运行的会话上设 置追踪	304
攻略 9-1 显示查询的执行计划	256	攻略 10-22 登录之后启用会话追踪	305
攻略 9-2 定制执行计划输出	258	攻略 10-23 追踪优化器的执行路径	306
攻略 9-3 图形化显示执行计划	261	攻略 10-24 生成 Oracle 错误自动追踪	309
攻略 9-4 解读一份执行计划	262	攻略 10-25 追踪后台进程	310
攻略 9-5 监控运行时间较长的 SQL 语句	264	攻略 10-26 启用 Oracle 监听器追踪	311
攻略 9-6 确定当前正在执行的耗占资源 的 SQL 语句	265	攻略 10-27 为数据卫士设置归档追踪	312
攻略 9-7 查看当前正在运行的 SQL 语 句的统计信息	266	第 11 章 SQL 自动调优	314
攻略 9-8 监控一个 SQL 执行计划的处 理过程	269	攻略 11-1 显示自动 SQL 调优工作详细 信息	316
攻略 9-9 确定过去执行的 SQL 语句中 最耗占资源的语句	271	攻略 11-2 显示 SQL 自动调优建议	318
攻略 9-10 比较系统修改后的 SQL 性能	273	攻略 11-3 生成 SQL 脚本来实现 自动调优建议	322
第 10 章 追踪 SQL 执行	278	攻略 11-4 修改 SQL 自动调优特性	323
攻略 10-1 环境准备	278	攻略 11-5 禁用和启用 SQL 自动调优	325
攻略 10-2 追踪一个特定的 SQL 语句	280	攻略 11-6 修改维护窗口属性	326
攻略 10-3 在你所拥有的会话中启用 追踪	282	攻略 11-7 创建 SQL 调优集对象	327
攻略 10-4 找到追踪文件	283	攻略 11-8 查看 AWR 中的资源密集型 SQL 语句	328
攻略 10-5 检查原始 SQL 追踪文件	284	攻略 11-9 查看内存中的资源 密集型 SQL 语句	330
攻略 10-6 分析 Oracle 追踪文件	285	攻略 11-10 用 AWR 中高资源消耗的 SQL 来填充优化集	332
攻略 10-7 使用 TKPROF 设置追踪文件的 格式	286	攻略 11-11 用内存中高资源消耗的 SQL 来填充调优集	333
攻略 10-8 分析 TKPROF 输出	287	攻略 11-12 将内存中所有 SQL 语句填充到 SQL 调优集	334
攻略 10-9 使用 Oracle 追踪分析器分析 追踪文件	290	攻略 11-13 显示 SQL 调优集的内容	335
攻略 10-10 追踪一个并行查询	293	攻略 11-14 有选择地从 SQL 调优集中删 除语句	337
攻略 10-11 追踪特定的并行查询进程	294	攻略 11-15 传输 SQL 调优集	338
		攻略 11-16 创建调优任务	340

攻略 11-17	手工运行 SQL 调优顾问	342
攻略 11-18	从数据库自动诊断监视器中 获得 SQL 调优建议	345
第 12 章	执行计划优化与一致性	348
攻略 12-1	创建并接受 SQL 概要	351
攻略 12-2	自动接受 SQL 概要文件	354
攻略 12-3	显示 SQL 概要文件信息	356
攻略 12-4	禁用 SQL 概要文件	358
攻略 12-5	删除 SQL 概要文件	360
攻略 12-6	移动 SQL 概要文件	361
攻略 12-7	自动增加计划基线	363
攻略 12-8	为一条 SQL 语句创建计划 基线	365
攻略 12-9	为包含在 SQL 调优集中的 SQL 语句创建计划基线	366
攻略 12-10	修改计划基线	368
攻略 12-11	确认是否存在计划基线	370
攻略 12-12	显示计划基线执行计划	371
攻略 12-13	在计划基线中加入一个新的 计划 (扩展)	372
攻略 12-14	禁用计划基线	375
攻略 12-15	移除计划基线信息	376
攻略 12-16	迁移计划基线	377
第 13 章	优化器配置	380
攻略 13-1	选择优化器目标	380
攻略 13-2	启用统计信息自动收集	381
攻略 13-3	为统计信息收集设置首选参 数	383
攻略 13-4	手工生成统计信息	388
攻略 13-5	锁定统计信息	389
攻略 13-6	处理统计信息的缺失	390
攻略 13-7	导出统计信息	392
攻略 13-8	还原以前版本的统计信息	393
攻略 13-9	收集系统统计信息	394
攻略 13-10	验证新的统计信息	397
攻略 13-11	强制优化器使用某个索引	399
攻略 13-12	启用查询优化器特性	400
攻略 13-13	阻止数据库创建柱状图	402
攻略 13-14	不使用绑定变量提高性能	403
攻略 13-15	理解自适应游标共享	406
攻略 13-16	在表达式上创建统计信息	411
攻略 13-17	为相关列创建统计信息	412
攻略 13-18	自动创建列组	413
攻略 13-19	维护分区表统计信息	415
攻略 13-20	为大表并行收集统计信息	416
第 14 章	实现查询提示	419
攻略 14-1	编写一个提示	419
攻略 14-2	改变访问路径	420
攻略 14-3	改变连接顺序	424
攻略 14-4	改变连接方法	425
攻略 14-5	改变优化器版本	428
攻略 14-6	在快速响应和整体优化之间 进行选择	428
攻略 14-7	进行直接路径插入	430
攻略 14-8	在视图加入提示	432
攻略 14-9	缓存查询结果	434
攻略 14-10	将分布式查询引导到一个 特定的数据库	437
攻略 14-11	收集查询执行的扩展 统计信息	441
攻略 14-12	启用查询改写	442
攻略 14-13	提升星型架构查询的性能	444
第 15 章	并行执行 SQL	447
攻略 15-1	为特定查询启用并行	447
攻略 15-2	在创建对象时启用并行	451
攻略 15-3	为已经存在的对象启用并行	452
攻略 15-4	实现并行 DML	453
攻略 15-5	并行创建表	456
攻略 15-6	并行创建索引	458
攻略 15-7	并行重建索引	459
攻略 15-8	并行移动分区	460
攻略 15-9	并行拆分分区	461
攻略 15-10	启用自动并行度	462
攻略 15-11	检查并行解释计划	463
攻略 15-12	监控并行操作	466
攻略 15-13	找出并行进程中的瓶颈	468
攻略 15-14	获取并行会话的详细信息	469
索引		471

第 1 章

优化表性能



本章着眼于影响表中数据存取性能的数据库特性。表的性能部分取决于在创建表之前所应用的数据库特性。例如，在最初创建数据库时所采用的物理存储特性以及相关的表空间都会在后来影响表的性能。类似地，表性能还受到最开始选择的物理特性的影响，例如表类型和数据类型。因此应用实践中使用的数据库、表空间和表的创建标准（并将性能问题放在心上），就形成了优化数据可用性和可扩展性的基础。

组成Oracle数据库的物理结构用来存储、管理、保护以及读取数据。在创建数据库的时候，可以选择应用一些与性能相关的特性。例如，数据文件的初始布局以及表空间的管理类型，都是在创建数据库时指定的。这时所实现的架构上的决策，通常都会产生很长远的影响。

表空间是支持管理一组数据文件的逻辑结构。数据文件就是磁盘上的物理文件。配置表空间时，要注意一些对性能会产生深远影响的特性，也就是本地管理表空间以及自动段存储管理表空间。如果合理地设置了这些特性，将来也就能最大限度地获得可接受的表性能。

表是数据库中存储数据的对象。数据库性能衡量的是应用能够以什么样的速度插入、更新、删除和查询数据。因此，本书就从优化表性能的攻略讲起。

本章首先介绍创建数据库和表空间时，可能会影响表性能的各方面因素。然后，讨论另外一些主题，比如根据与性能相关的业务需求，选择表类型和数据类型。稍后介绍的主题包括管理表空间使用情况的物理实现方式。本章还会详细介绍其他问题，例如探测表碎片、处理位于高水位线下方的空闲空间、行链接，以及数据压缩。除此以外还会描述Oracle段顾问（Oracle Segment Advisor）。这个工具很好用，能够帮助你自动探测并解决表碎片和未使用的空间问题。

攻略 1-1 创建具有最优性能的数据库

问题描述

你认识到，最初创建数据库时，有一些属性（如果启用了的话）将会对数据库中表的性能和可用性产生长远的影响。具体来说，创建数据库时，需要做下面这些事情。

- ❑ 强制数据库中创建的每一个表空间都必须是本地管理的。本地管理表空间能比已经被弃用的字典管理技术提供更好的性能。

- ❑ 确保数据库为每个用户自动分配一个默认的永久表空间。这可以保证在创建用户时，自动分配一个默认的表空间，而不是SYSTEM系统表空间。你不能让用户总是将对象建在SYSTEM表空间中，因为这样会对性能和可用性产生负面影响。
- ❑ 确保数据库会为每个用户自动分配一个默认的临时表空间。这可以保证在创建用户时，自动分配一个临时表空间，而不是SYSTEM表空间。你不能总是让用户使用SYSTEM表空间作为排序操作运算的临时表空间，因为这样会对性能和可用性产生负面影响。

解决方案

使用类似下面这样的脚本创建遵循合理标准的数据库，这就为数据库的高性能打下了良好的基础：

```
CREATE DATABASE O11R2
  MAXLOGFILES 16
  MAXLOGMEMBERS 4
  MAXDATAFILES 1024
  MAXINSTANCES 1
  MAXLOGHISTORY 680
  CHARACTER SET AL32UTF8
DATAFILE
  '/ora01/dbfile/O11R2/system01.dbf'
  SIZE 500M REUSE
  EXTENT MANAGEMENT LOCAL
UNDO TABLESPACE undotbs1 DATAFILE
  '/ora02/dbfile/O11R2/undotbs01.dbf'
  SIZE 800M
SYSaux DATAFILE
  '/ora03/dbfile/O11R2/sysaux01.dbf'
  SIZE 500M
DEFAULT TEMPORARY TABLESPACE TEMP TEMPFILE
  '/ora02/dbfile/O11R2/temp01.dbf'
  SIZE 500M
DEFAULT TABLESPACE USERS DATAFILE
  '/ora01/dbfile/O11R2/users01.dbf'
  SIZE 50M
LOGFILE GROUP 1
  ('/ora01/oraredo/O11R2/redo01a.rdo',
   '/ora02/oraredo/O11R2/redo01b.rdo') SIZE 200M,
GROUP 2
  ('/ora01/oraredo/O11R2/redo02a.rdo',
   '/ora02/oraredo/O11R2/redo02b.rdo') SIZE 200M,
GROUP 3
  ('/ora01/oraredo/O11R2/redo03a.rdo',
   '/ora02/oraredo/O11R2/redo03b.rdo') SIZE 200M
USER sys IDENTIFIED BY topfoo
USER system IDENTIFIED BY topsecretfoo;
```

上面这段CREATE DATABASE脚本通过实现下列功能，为数据库的性能打下良好基础。

- ❑ 通过EXTENT MANAGEMENT LOCAL子句，将SYSTEM表空间定义为本地管理表空间。这确保了数据库中创建的所有表空间都是本地管理的。在Oracle Database 11g R2或更高版本中，EXTENT MANAGEMENT DICTIONARY子句已经被弃用了。
- ❑ 为创建时没有显式定义默认表空间的用户，定义一个默认的表空间USERS。这可以防止为

用户分配SYSTEM系统表空间作为默认表空间。创建将SYSTEM作为默认表空间的用户，对性能有负面影响。

- ❑ 为所有用户定义一个默认临时表空间，名为TEMP。这可以防止为用户分配SYSTEM表空间作为默认的临时表空间。创建将SYSTEM作为默认临时表空间的用户，对性能有负面影响，因为这会造成SYSTEM表空间中的资源争夺。

良好的性能始于正确的数据库配置。以上推荐的这些配置有助于为表数据创建一个可靠的基础构造。

工作原理

合理地配置并创建数据库有助于保证数据库的优良性能。尽管确实可以在创建数据库之后再修改其功能，但是，通常的情况是，写得很差的CREATE DATABASE脚本将会对数据库性能产生永久的影响。生产环境中，数据库有时很难有机会停机，对配置不恰当的地方进行重新配置。如果可能的话，最好是从创建数据库开始，在创建环境的每一步都仔细考虑性能问题。

创建数据库时，还需要考虑影响可维护性的功能。容易维护的数据库能正常运行更长时间，而这也正是总体性能的一个重要部分。“解决方案”部分的CREATE DATABASE语句同时还考虑了下面这些可维护性方面的特性。

- ❑ 创建一个自动的UNDO表空间（自动撤销管理通过设置UNDO_MANAGEMENT和UNDO_TABLESPACE初始化参数来启用）。这使Oracle可以自动管理回滚段，你也就不必定期进行监控和微调了。
- ❑ 按照环境中的特定标准，将数据文件放到相应文件夹中。这有助于维护和管理，从而使数据库具备更好的长期可用性，从而获得更好的性能。
- ❑ 将数据库管理员（DBA）相关用户的密码设置为非默认值。这可以进一步确保数据库的安全性，从长远来看也会影响到性能（如果有攻击者入侵数据库并删除数据，那么性能也将会受到损害）。
- ❑ 创建三组联机重做日志文件，每组两个成员，大小适合事务加载。重做日志文件的大小直接影响切换的频率。如果重做日志切换太过频繁，就会导致性能下降。

你应该花点时间确认每一个数据库都是按照广泛认可的标准创建的，这有助于保证数据库具有坚实的性能基础。

如果你接手了一个数据库，并且想检查一下默认的永久表空间设置，可以使用类似下面的查询：

```
SELECT *  
FROM database_properties  
WHERE property_name = 'DEFAULT_PERMANENT_TABLESPACE';
```

如果你需要修改默认的永久表空间，按照下面这样来做：

```
SQL> alter database default tablespace users;
```

要检验默认临时表空间的设置，使用这个查询：

```
SELECT *  
FROM database_properties  
WHERE property_name = 'DEFAULT_TEMP_TABLESPACE';
```

要修改临时表空间的设置，可以像下面这样做：

```
SQL> alter database default temporary tablespace temp;
```

可以通过下面这个查询来检验UNDO表空间的设置：

```
select name, value
from v$parameter
where name in ('undo_management', 'undo_tablespace');
```

如果你需要修改重做(undo)表空间，首先创建一个新的重做表空间，然后使用ALTER SYSTEM SET UNDO_TABLESPACE语句。

攻略 1-2 创建具有最优性能的表空间

问题描述

你已经认识到表空间是存储数据库对象(例如表和索引)的逻辑容器。进一步说，你已经知道如果在创建对象时不为其指定存储属性，那么相应的表和索引就将会自动继承表空间(这些表和索引在其中创建)的存储特性。因此就需要以能够使得表性能和可维护性最优化的方法来创建表空间。

解决方案

如果可以选择，所创建的表空间都应该具备以下两个功能：

□ 本地管理；

□ ASSM (Automatic segment space management, 自动段空间管理)

下面这个例子创建了包含上述两个功能的表空间：

```
create tablespace tools
datafile 'ora01/dbfile/INVREP/tools01.dbf'
size 100m -- Fixed datafile size
extent management local -- Locally managed
uniform size 128k -- Uniform extent size
segment space management auto -- ASSM
/
```

注意 自Oracle Database 11g R2版本起，EXTENT MANAGEMENT DICTIONARY子句已经被弃用。

本地管理表空间比字典管理表空间的效率更高。这个功能可以通过EXTENT MANAGEMENT LOCAL子句来启用。此外，如果在创建数据库时，将SYSTEM表空间设置为本地管理，那么之后将不允许创建字典管理的表空间。这正是我们想要的。

利用ASSM功能，Oracle能够自动管理很多存储特性，而以前这些特性需要由DBA逐个表手工进行设置。ASSM通过SEGMENT SPACE MANAGEMENT AUTO子句来启用。使用ASSM，就可以从这些手工的数据库微调操作中解放出来。并且，Oracle的一些空间管理功能(例如收缩表和安全文件大对象)只有在使用ASSM时才允许使用。如果你想利用这些功能，就必须创建使用ASSM的表空间。

可以通过UNIFORM SIZE子句，将表空间中每一次空间扩展的大小设置为同样的值。或者也可