



ORACLE PRESS™ — EXCLUSIVELY FROM MCGRAW-HILL/OSBORNE

Oracle Wait Interface:
A Practical Guide to Performance
Diagnostics & Tuning

Oracle Wait Interface

性能诊断与调整实践指南



ORIGINAL • AUTHENTIC

Oracle Press™

ONLY FROM OSBORNE

Richmond Shee
(美) Kirtikumar Deshpande 著
K Gopalakrishnan
高 猛 江仁容 译

**Mc
Graw
Hill** Education



清华大学出版社



Oracle Wait Interface:
A Practical Guide to Performance
Diagnosis & Tuning

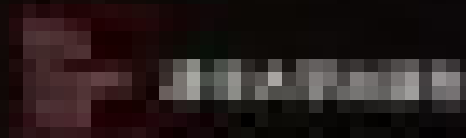
Oracle Wait Interface

性能诊断与调整实践指南



Oracle
Performance
Expert's Voice
Series

Written and Edited by
Michael G. Christensen
© Copyright 2012
ISBN 978-1-56582-712-5



Oracle Wait Interface

性能诊断与调整实践指南

Richmond Shee

(美) Kirtikumar Deshpande 著

K Gopalakrishnan

高猛 江仁容 译

清华大学出版社

北 京

Richmond Shee, Kirtikumar Deshpande, K Gopalakrishnan

Oracle Wait Interface: A Practical Guide to Performance Diagnostics & Tuning

EISBN: 0-07-222729-X

Copyright © 2004 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 by any means, or stored in a database or retrieval system, without the prior written permission of the publisher.

Simplified Chinese translation edition is published and distributed exclusively by Tsinghua University Press under the authorization by McGraw-Hill Education(Asia) Co., within the territory of the People's Republic of China only (excluding Hong Kong, Macao SAR and Taiwan). Unauthorized export of this edition is a violation of the Copyright Act. Violation of this Law is subject to Civil and Criminal Penalties.

本书中文简体字翻译版由美国麦格劳-希尔教育出版(亚洲)公司授权清华大学出版社在中华人民共和国境内(不包括中国香港、澳门特别行政区和中国台湾地区)独家出版发行。未经许可之出口视为违反著作权法,将受法律之制裁。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。

北京市版权局著作权合同登记号 图字: 01-2004-3721

版权所有, 翻印必究。举报电话: 010-62782989 13501256678 13801310933

本书封面贴有 McGraw-Hill 公司防伪标签, 无标签者不得销售。

图书在版编目(CIP)数据

Oracle Wait Interface 性能诊断与调整实践指南/(美)西(Shee, R.), (美)德旁特(Deshpande, K.), (美)戈帕拉克里纳(Gopalakrishnan, K.)著; 高猛, 江仁容译. —北京: 清华大学出版社, 2005.7

书名原文: Oracle Wait Interface: A Practical Guide to Performance Diagnostics & Tuning

ISBN 7-302-10987-7

I. O… II. ①西… ②德… ③戈… ④高… ⑤江… III. 关系数据库—数据库管理系统, Oracle IV. TP311.138

中国版本图书馆 CIP 数据核字(2005)第 048937 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: 于 平

封面设计: 康 博

版式设计: 康 博

印装者: 北京嘉实印刷有限公司

发行者: 新华书店总店北京发行所

开本: 185 × 260 印张: 17 字数: 435 千字

版次: 2005 年 7 月第 1 版 2005 年 7 月第 1 次印刷

书号: ISBN 7-302-10987-7/TP · 7286

印数: 1 ~ 3000

定价: 39.00 元



前 言

商业全球化创造了新的竞争前景，最先占领市场的推动力往往会违反常规以及已证明的应用程序和数据库开发范例。为了获得竞争优势，各个公司利用了最新的计算机硬件和存储技术，从而获得并分析关于客户和支出形态的大量数据。这就给大型数据库、复杂的多层式应用程序带来了一定的风险，更不用说可能产生性能问题。数据库管理员们面临前所未有的故障检修和解决性能问题的挑战，这些问题往往都贴上了“数据库问题”的标签。

在 Oracle 7.0.12 以前的版本中，Oracle 公司建议用基于比率的方法测量数据库性能。作为结果，人们开发出了基于各种比率的模型，以此来测量数据库性能。人们已经很长时间学习和使用该模型，而且仍然有一些 DBA(数据库管理员)在不断地实践这个模型。然而，事实证明，该模型并不适合于查找系统中真正的瓶颈问题，更不用说寻找解决方案。尽管有各种各样的比率，然而基于比率的方法却不能回答如下的问题：是什么原因造成应用程序运行缓慢？

在 Oracle 7.0.12 版本中，Oracle 公司引进了基于等待的方法(即 Oracle Wait Interface, OWI)来分析系统的性能问题。第一次，DBA 可以发现进程在不同的资源上花费了多少时间，以及何种资源是系统中的瓶颈。等待时间越长，进程的响应时间越慢。现在，通过将缓慢的性能与瓶颈联系起来的方法，DBA 可以更容易地标识主要的瓶颈，无论它是否与应用程序、数据库或网络相关。不再需要推测工作。然而，许多 DBA 都很晚才了解 OWI 的相关知识。起先，只有少数 Oracle 公司的员工知道基于等待的性能诊断知识和正确用法。其他人可用的信息相当少。从那时起，Oracle 公司在技术和知识传输方法方面对 OWI 都进行了实质性的改进。事实证明，对于正确标识系统瓶颈和找出正确解决方案，OWI 都是极其可靠和安全的模型。Oracle

RDBMS 8、8i、9i 和 10g 对 Oracle 内核进行了相当可观的改进，以报告等待时间，基于等待的模型正在非常快速地成熟起来。

在过去的 10 年中，已经有一些作者编写了关于调整和改进数据库性能的方法和工具。其范围从 SQL 调整到数据库内部构件。遗憾的是，其中的 Wait Interface 模型的内容未及时得以详细说明和补充。本书将努力填补这个空白。

本书不是指导如何调整 SQL 语句或 PL/SQL 程序，而是帮助您标识可能需要调整的 SQL 语句和数据库结构。本书面向所有层次的 Oracle DBA。我们相信，只要阅读了本书，您就会开始使用 OWI 诊断性能问题。如果是这样，我们编写本书的目的就达到了。

我们愿意倾听您的成功故事和建议，以及各种批评和意见。学无止境。我们决不会尝试覆盖所有可能的性能调整。我们认为这仅仅是开始。您可以通过电子邮件联系作者：richmondshee@yahoo.com、Kirtikumar_Deshpande@yahoo.com 和 kaygopal@yahoo.com。

如何使用本书

极少有人会一次性从头至尾地阅读技术书籍。技术书籍通常是用作参考书籍。然而，我们建议您读完本书的所有章节和附录。无论您在 Oracle Database 调整 and 性能故障查找中的经验水平如何，您一定会在每一章中发现一些新的内容。

本书共有 9 章和 5 个附录。下面将分别介绍。

第 1 章

第 1 章介绍了 Oracle Wait Interface，为了解学习和使用 OWI 的原因打下基础。本章解释了在需要诊断缓慢的应用程序响应时间时，为什么基于命中率(hit-ratio)的方法不起作用。

第 2 章

第 2 章详细地讨论了 OWI 组件。定义了 Oracle 等待事件，描述了各种 OWI 视图及其应用，还解释了如何使用扩展的 SQL 追踪。我们将介绍在何处、何时以及如何查找等待事件信息。

第 3 章

第 3 章介绍了最常见的 Oracle 等待事件。本章详细描述了这些事件，并显示了 Oracle 提供的、用于分析这些事件的信息。此外，本章还讨论了如何追踪 CPU 使用状况统计信息。

第 4 章

第 4 章解释了为什么监控和收集会话级等待事件如此重要的原因，讨论了性能历史数据的重要性。本章描述了如何将简单的数据收集工具组合起来以捕获和存储会话级 Oracle 等待事件数据。

第 5 章

第 5 章讨论了如何诊断和解决最常见的与 I/O 相关的等待事件的问题。讨论的事件包括 db file sequential read、db file scattered read、direct path read、direct path write、log file parallel write、db file parallel write 和 control file parallel write。

第 6 章

第 6 章深入讨论了 latch free、enqueue 和 buffer busy waits 等待事件。我们将全面介绍这些等待事件的相关内容。本章讨论了锁存器和排队的区别，解释了 Oracle 如何串行化内存结构的访问。本章将介绍如何诊断和解决与锁及串行化有关的问题。关于这些事件的信息，我们相信，目前还没有其他出版物中能像本章介绍得这么全面。

第 7 章

第 7 章详细介绍了与等待时间有关的等待事件。我们将介绍如何诊断和解决与 log file sync、log buffer space、free buffer waits、write complete waits、log file switch completion 和 log file switch(checkpoint incomplete)等待事件有关的问题。

第 8 章

第 8 章解释了在 Oracle Real Application Clusters 环境中缓冲存储器管理的工作方式，讨论了 RAC 环境中常见的 Oracle 等待事件。我们将介绍如何诊断和解决与全局高速缓存等待事件有关的问题。

第 9 章

第 9 章介绍了 Oracle Database 10g 中新增的自动化性能监控和诊断功能。本章讨论了 Oracle Database 10g 收集的不同类型的数据库统计信息。我们将介绍 Oracle 如何收集并分析性能统计数据，用推荐和“顾问”方案提供补救解决方案。我们还会介绍 Automatic Workload Repository、Active Session History、Automatic Database Diagnostic Monitor 的工作方式。我们将用一系列 Oracle Enterprise Manager 屏幕截图来展示找到性能问题根本原因的工作是多么轻松。

附录 A

附录 A 讨论了 Oracle 诊断事件。这些事件不同于 Oracle 等待事件。我们将介绍如何以及何时使用这些诊断事件。

附录 B

附录 B 列出了 Oracle Database 10g 中的所有排队等待事件。

附录 C

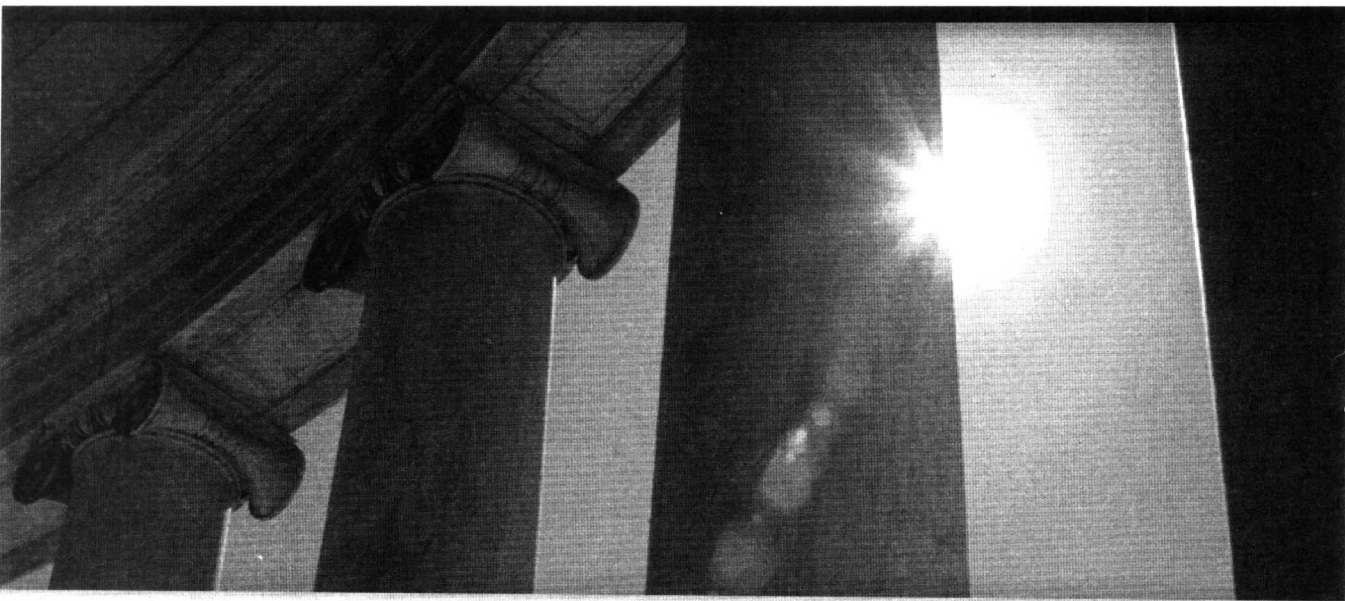
附录 C 介绍了用于为各种数据或内存转储创建追踪文件的工具和过程。通常 Oracle 技术支持部门需要这些追踪文件和转储，以诊断错误的根本原因，如 ORA-0600 或 ORA-7445。

附录 D

附录 D 描述了通过非 SQL 接口(如 C 语言程序)直接访问 Oracle SGA 的方法。这样的方法是采样 SGA 中与性能有关的信息的最快方法，无需引进额外的解析、锁存等开销。

附录 E

附录 E 列出了本书用到的参考资料。

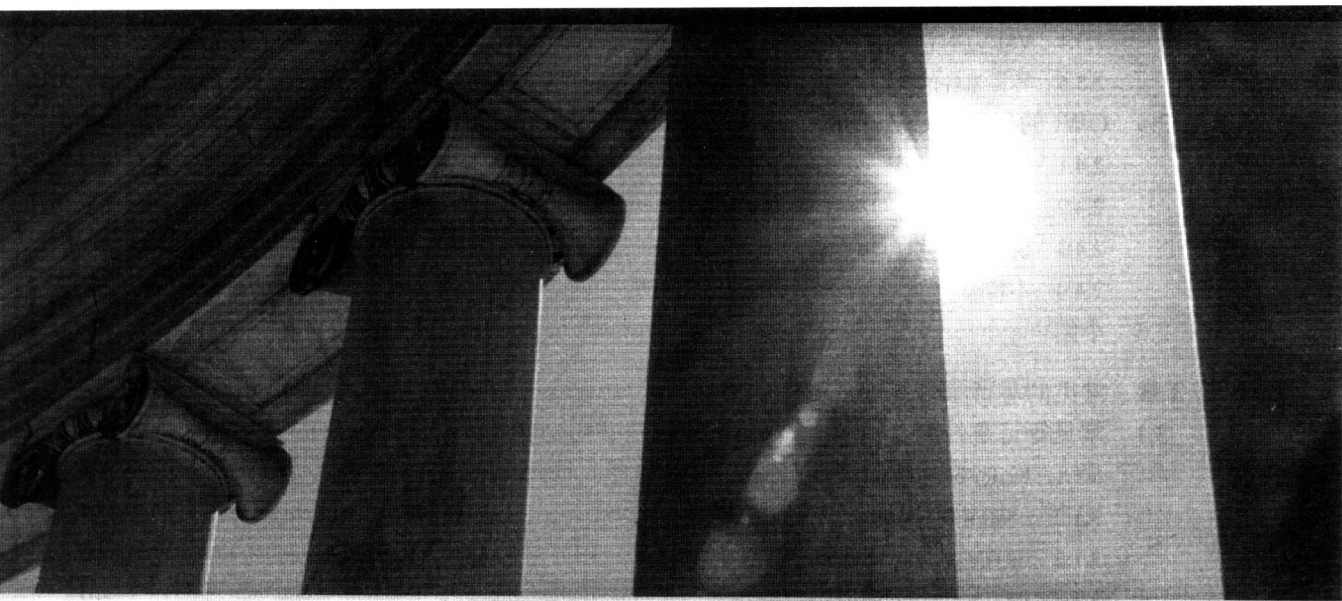


作者简介

Richmond Shee 是全球综合性通信公司 Sprint(www.sprint.com)的高级数据库架构师。Richmond 自 1984 年开始从事关系数据库工作。他是 DBA(数据库管理员)的顾问,并帮助确定适合在整个公司实现 Oracle RDBMS 技术的方针。Richmond 负责 Sprint 公司所有最关键数据库的调整工作。在他的众多成就中,最突出的是在 Sprint 公司倡导了 Oracle Wait Interface 的使用。他还发明了基于等待的性能数据收集器,这项技术正在申请专利中。他是国际 Oracle 用户组(International Oracle Users Group)的知名主席,经常在 Kansas City OUG 中演讲。Richmond 的电子邮件地址为 richmondshee@yahoo.com。

Kirtikumar Deshpande(Kirti)在信息技术领域方面已有 25 年的丰富经验,其中有十几年担任 Oracle 数据库管理员。他拥有理学士(物理)学位和工程学士(生物医学)学位。他与其他人合著了一本书 *Oracle Performance Tuning 101*, 由 Oracle Press 在 2001 年 5 月出版。他在当地 Oracle 用户组非正式会议,以及国内、国际 Oracle 用户组会议上发表多篇论文。目前他是 Verizon Information Services(www.superpages.com)的高级 Oracle 数据库管理员。Kirti 的电子邮件地址为 Kirtikumar_Deshpande@yahoo.com。

K Gopalakrishnan(Gopal)是 Oracle Solution Services(印度)的首席顾问,专攻性能调整、高可用性和灾难恢复。他是公认的 Oracle RAC and Database Internals 专家。Gopal 根据其丰富的专业经验已为解决全世界电信巨头、银行和大学中解决许多令人头疼的性能问题作出贡献。他也是一位多产作家,经常在 *Oracle Internals* 杂志上发表文章。他拥有印度 Madras 大学的计算机科学和工程学位,以及 8 年多的从业经验。Gopal 的电子邮件地址为 kaygopal@yahoo.com。



目 录

第 1 章	介绍 Oracle Wait Interface	1
1.1	Oracle 性能优化的旧方式	2
1.2	高速缓存命中率极度无效率的原因	2
1.3	Oracle 性能优化的新方式	4
1.4	OWI 基本原理	4
1.5	数据库响应时间调整模型	6
1.6	思维的变迁	8
1.7	小结	9
第 2 章	Oracle Wait Interface 组件	11
2.1	什么是等待事件	11
2.2	OWI 组件	12
2.2.1	V\$EVENT_NAME 视图	13
2.2.2	V\$SYSTEM_EVENT 视图	14
2.2.3	V\$SESSION_EVENT 视图	17
2.2.4	V\$SESSION_WAIT 视图	19
2.2.5	追踪事件 10046—— 扩充的 SQL 追踪	21
2.3	Oracle Database 10g Release 1 中的新 OWI 视图	27
2.3.1	V\$SESSION_WAIT_HISTORY 视图	27
2.3.2	V\$SYSTEM_WAIT_CLASS 视图	28

2.3.3	V\$SESSION_WAIT_CLASS 视图	29
2.3.4	V\$EVENT_HISTOGRAM 视图	29
2.3.5	等待事件的类型	31
2.4	OWI 的局限	33
2.4.1	没有 CPU 统计	33
2.4.2	没有端到端的可视性	33
2.4.3	没有历史数据	34
2.4.4	不精确	34
2.5	小结	35
第 3 章	常见的等待事件	37
3.1	常见等待事件简介	37
3.1.1	buffer busy waits 等待事件	38
3.1.2	control file parallel write 等待事件	39
3.1.3	db file parallel read 等待事件	40
3.1.4	db file parallel write 等待事件	40
3.1.5	db file scattered read 等待事件	41
3.1.6	db file sequential read 等待事件	41
3.1.7	db file single write 等待事件	42
3.1.8	direct path read 等待事件	42
3.1.9	direct path write 等待事件	43
3.1.10	enqueue 等待事件	43
3.1.11	free buffer waits 等待事件	44
3.1.12	latch free 等待事件	45
3.1.13	library cache pin 等待事件	46
3.1.14	library cache lock 等待事件	46
3.1.15	log buffer space 等待事件	47
3.1.16	log file parallel write 等待事件	47
3.1.17	log file sequential read 等待事件	47
3.1.18	log file switch(archiving needed)等待事件	48
3.1.19	log file switch(checkpoint incomplete)等待事件	48
3.1.20	log file switch completion 等待事件	48
3.1.21	log file sync 等待事件	49
3.1.22	SQL*Net message from client 等待事件	49
3.1.23	SQL*Net message to client 等待事件	50
3.2	Oracle Real Application Clusters 环境中的常见等待事件	50
3.2.1	global cache cr request	50
3.2.2	buffer busy global cache 等待事件	51
3.2.3	buffer busy global cr 等待事件	52
3.2.4	global cache busy 等待事件	52

3.2.5	global cache null to x 等待事件	53
3.2.6	global cache null to s 等待事件	53
3.2.7	global cache s to x 等待事件	54
3.2.8	global cache open x 等待事件	54
3.2.9	global cache open s 等待事件	54
3.2.10	row cache lock	55
3.3	追踪 CPU 和其他统计	55
3.4	小结	57
第 4 章	OWI 监控和收集方法	59
4.1	性能历史数据重要的原因	60
4.2	对根本原因快速而精确的分析	60
4.3	追踪事件 10046 不适合用作数据收集器的原因	61
4.4	Statspack 不适合用作数据收集器的原因	62
4.5	将 Database Logoff Trigger 用作数据收集器	62
4.6	利用 PL/SQL 过程进行性能数据采样	66
4.6.1	数据源	66
4.6.2	采样频率	67
4.6.3	存储库	69
4.6.4	监控的事件	70
4.6.5	优点与缺点	76
4.7	利用非 SQL 式 SGA 访问进行性能数据采样	77
4.8	小结	77
第 5 章	解释常见的与 I/O 有关的等待事件	79
5.1	db file sequential read 等待事件	80
5.2	db file scattered read 等待事件	86
5.2.1	常见的原因、诊断和动作	86
5.2.2	db file sequential read 事件出现在全表扫描操作中的原因	89
5.2.3	全局扫描操作比 MBRC 请求更少块的原因	90
5.2.4	设置 DB_FILE_MULTIBLOCK_READ_COUNT(MBRC)	91
5.2.5	为什么物理 I/O 需要巨大代价	92
5.3	direct path read 事件	92
5.3.1	常见的原因、诊断和动作	92
5.3.2	关注的初始参数	96
5.4	direct path write 等待事件	98
5.5	db file parallel write 等待事件	100
5.6	log file parallel write 等待事件	103
5.7	control file parallel write 等待事件	106
5.8	小结	107

第 6 章 解释与锁有关的等待事件	109
6.1 latch free 等待事件	110
6.1.1 什么是锁存器	110
6.1.2 锁存器和锁两者之间的区别	110
6.1.3 锁存器家族	111
6.1.4 锁存器获取	111
6.1.5 锁存器分类	112
6.1.6 latch free 等待事件表明的内容	114
6.1.7 锁存器丢失位置	115
6.1.8 Oracle Database 10g Release 1 中的锁存器	115
6.1.9 常见的原因、诊断和动作	116
6.1.10 shared pool 锁存器和 library cache 锁存器	117
6.1.11 cache buffers chains 锁存器	123
6.1.12 cache buffers lru chain 锁存器	129
6.1.13 row cache objects 锁存器	131
6.2 enqueue 等待事件	131
6.2.1 什么是排队	132
6.2.2 什么是排队资源	132
6.2.3 什么是排队锁	134
6.2.4 排队体系结构	135
6.2.5 解码排队类型和模式	135
6.2.6 常见的原因、诊断和动作	137
6.3 buffer busy waits 等待事件	143
6.4 小结	150
第 7 章 解释与延迟有关的常见等待事件	153
7.1 log file sync 等待事件	153
7.2 log buffer space 等待事件	158
7.3 free buffer waits 等待事件	159
7.4 write complete waits 等待事件	163
7.5 log file switch completion 等待事件	163
7.6 log file switch(checkpoint incomplete)等待事件	164
7.7 小结	165
第 8 章 Real Application Clusters 环境中的等待事件	167
8.1 Real Application Clusters 中等待事件的特殊之处	167
8.2 全局高速缓存等待事件	173
8.2.1 global cache cr request 等待事件	173
8.2.2 global cache busy 等待事件	176
8.3 Oracle Database 10g 中增加的 RAC 等待事件	178
8.4 enqueue 等待事件	178
8.5 小结	182

第 9 章 Oracle Database 10g 中的性能管理	183
9.1 数据库统计	184
9.1.1 时间模型统计	184
9.1.2 等待模型统计	185
9.1.3 操作系统统计	185
9.1.4 其他 SQL 统计	186
9.1.5 数据库度量	186
9.2 新的后台进程	186
9.3 Automatic Workload Repository	187
9.3.1 存储库快照	188
9.3.2 快照基线	188
9.3.3 使用 EM 管理 AWR	188
9.3.4 手工管理 AWR	191
9.4 Active Session History	194
9.4.1 什么是活动会话	195
9.4.2 ASH 的组成部分	195
9.5 ADDM	199
9.5.1 ADDM 设置	200
9.5.2 使用 EM 访问 ADDM	201
9.5.3 手工运行 ADDM 报告	207
9.5.4 ADDM 视图	208
9.6 小结	208
附录 A Oracle Database 10g 的诊断事件	209
A.1 Oracle 诊断事件	209
A.1.1 诊断事件的类型	210
A.1.2 设置诊断事件	212
A.1.3 事件的内部工作	216
附录 B Oracle Database 10g 中的 enqueue 等待事件	219
附录 C Oracle 转储和追踪	227
C.1 oradebug: 追踪和转储的重要工具	227
C.2 数据块转储	229
C.3 缓冲区转储	230
C.3.1 语法	230
C.3.2 控制转储信息	231
C.4 缓冲区转储	231
C.4.1 语法	231
C.4.2 使用级别控制转储信息	232
C.5 文件头转储	232

C.5.1	语法	232
C.5.2	使用级别控制转储信息	233
C.6	控制文件转储	234
C.6.1	语法	234
C.6.2	使用级别控制转储信息	234
C.7	堆转储	234
C.7.1	语法	235
C.7.2	使用级别控制转储信息	235
C.8	库高速缓存转储	236
C.8.1	语法	236
C.8.2	使用级别控制转储信息	236
C.9	进程状态转储	236
C.10	Shared Server 状态转储	237
C.10.1	语法	237
C.10.2	使用级别控制转储信息	238
C.11	系统状态转储	238
C.11.1	语法	238
C.11.2	使用级别控制转储信息	238
C.12	重做日志转储	239
C.12.1	语法	239
C.12.2	使用级别控制转储信息	239
附录 D	直接访问 SGA	241
D.1	开销	241
D.2	安全性	242
D.3	速度	242
D.4	并发性	242
D.5	获取隐藏信息	242
D.6	X\$视图简介	242
D.7	必要的成份	244
D.7.1	查找 SGA ID	245
D.7.2	查找 SGA 基地址	245
D.7.3	查找 X\$KSUSECST 的起始地址	246
D.7.4	查找 X\$KSUSECST 结构中的记录大小	246
D.7.5	查找 X\$KSUSECST 结构中的记录数	246
D.7.6	查找 X\$KSUSECST 视图列的偏移量	247
D.8	使用 C 程序访问 SGA	248
附录 E	参考文献	253



第 1 章

介绍 Oracle Wait Interface

速度是 21 世纪的流行词汇。以前人们都说，“死时拥有最多玩物的人是赢家”。如今是“死时拥有最快玩物的人是赢家”。现在无论是在个人生活中，还是在所从事的工作中，人们都要求速度。因此就产生了这些名称：快速轿车、快速计算机、快车道、快速 Internet 连接、快速减肥、快速治疗、快餐等。

商业公司竞争激烈，为的是率先将新产品推向市场。为了以速度抢占市场，人们往往不再使用过去常用的常规软件工程过程，包括诸如应用分析、设计、代码开发、测试和实现。现在，程序员在需求建立之前就开始编码，而需求在整个开发生命周期中始终在变化。此外，为满足不现实的产品发布日期，常常缩短甚至完全取消测试阶段。不用说，许多未经测试的应用软件在发布那天才与产品数据库相交互。

与当今快节奏的世界相适应，数据库的规模在不断扩大。这是因为商业公司发现了更长时间地、有时甚至是“永久地”保存历史数据的价值，这样可以进行各种分析从而赢得竞争优势。如果数据库可伸缩性不强，那么即使提供了性能更好而且更加可靠的硬件，处理时间也会比较长。

这些问题慢慢聚集，后来就衍变成所谓的“数据库性能”问题。遗憾的是，数据库管理员 (DBA) 不得不定期处理这些问题，并承担责任。公司 IT 部门希望 DBA 掌握所需的技术知识，

并能全天候地作好准备以快速解决所有性能问题。由于这个原因，性能监控和优化功能显得尤为突出，并耗费了 DBA 的大部分时间。DBA 需要可靠的性能监控和调整方法。Oracle 公司通过构建 Oracle Wait Interface(OWI)来满足这一需求，以帮助 DBA 快速有效地诊断性能问题。

本书介绍有关 Oracle Wait Interface 的内容。它为 DBA 提供了如何使用 OWI 的详细知识，以及基于大型复杂数据库安装实践经验的性能诊断和故障检修方法。DBA 将深入了解 OWI 方法的知识，并将能够运用这些知识解决性能问题。

本章将对 OWI 和许多 DBA 仍采用的基于比率的方法进行比较。我们将概述基于比率的方法的局限性，并强调 OWI 的优点，从而确定选择 OWI 方法的优势。我们真诚地希望本书对转而使用 OWI 作为主要的 Oracle 性能优化方法的 DBA 能有所帮助。

1.1 Oracle 性能优化的旧方式

有人说只有了解过去的生活之后，才能真正地认识现在所拥有的生活。这句话同样适用于 Oracle 性能优化领域。Oracle 的早期版本未提供标识性能瓶颈的可靠方法。性能优化是一项困难且复杂的任务。人人使用高速缓存命中率作为监控数据库性能的衡量标准。为了充分认识 OWI 调整方法，必须知道基于命中率调整方法的问题和局限性。对我们中的许多人来说，这是一次沿着记忆通道的旅行，而对那些不是在基于比率调整方法时代长大的人来说，他们可以了解前辈们的那段历史。

从 Oracle RDBMS 起，Oracle DBA 就开始通过观察一些比率数来调整数据库和实例。其基本思想是在可以接受的范围或限度内保持所有数据库元素的正常操作。一些值得注意的比率是缓冲存储器命中率、库高速缓存命中率/丢失率(Oracle 7.0)和锁存器获取率/丢失率。大家可能都不会忘记下面这些戒律：

- 必须保持缓冲存储器命中率高于 90%
- 数据字典丢失率必须一直低于 10%，库高速缓存不应等同于数据字典——它应有自身的比率
- SQL 区域 gethitratio 和 pinhitratio 也必须总是高于 90%。此外，重载到 pin 的比率必须小于等于 1%。如果比率不对，则应增加共享池的大小，但不能从缓冲存储器盗取内存。当为共享池添加内存时，同时也为缓冲存储器安排一些内存。这样将会提高高速缓存命中率
- willing-to-wait 锁存器命中率接近于 1。如果不是，则必须增加 SPIN_COUNT，但必须小心不要超过 CPU 的负荷

以及其他诸如此类的戒律。

从这些内容可以看出，旧方式太繁琐了，不容易被公众接受。我们可以使用更简单、更好的方式。

1.2 高速缓存命中率极度无效率的原因

命中率基本原理对 Oracle 数据库管理并不罕见。它用途广泛并深刻影响日常生活的许多方面。举个本地城市供水部门的例子。他们的目标是将饮用水的质量保持在环境保护署规定的限

度之下。现在，城市水质达到甚至远远超过设定的限度，但是这并不意味着每个市民都感到愉快。有些人对饮用水中的某些物质比一般人更容易过敏。如果您是这些不幸者中的一员，就需要自行解决这一问题。如果水质已经在可接受的限度之内，那么城市供水部门很可能就不会对此采取任何行动。

实际上，我们为数据库用户提供的就是这种客户服务。如果他们说他们的任务运行缓慢，则我们所做的全部事情就是确认所有比率在规定的范围内。这些比率中缓冲存储器命中率是关注的中心。如果比率在规定范围内，那么我们将用户打发走，并告诉他们实际上没有问题。如果他们继续抱怨，则我们将转向作业的 SQL 追踪，用 tkprof(Transient Kernel Profiler, 瞬态内核描述器)检查追踪文件，并调整一些 SQL 语句。我们还能干什么呢？所有有关性能调整的书籍和课程都教导我们如果比率在可接受的限度内，用户就应该满意。这种服务与那些只知道如何根据病人血压值进行治疗的医生没有区别。假设您割裂了手腕，失血很多，痛得打滚。而医生检查您的血压后，发现它在可接受的范围内，就会将您打发回家并要求您在血压偏低的时候再来。

如果比率数不在规定范围内，通常(或许也是惟一)的解决方案是添加更多的内存。有过这种经历的人无疑会记得这样充满痛苦的教训：更大的缓冲存储器并不总是带来更好的性能。还记得安装了添加的内存之后，您十指交叉，满怀希望并虔诚祈祷性能将得到提高吗？但性能并没有提高，反而降低了。您感到吃惊和难以置信，完全不能理解原因，更不知道还可以做些什么。您失去了信心，身处困境。在随后与客户开的会议上，有人建议，“为什么我们不求助于业界知名的性能调整专家呢？”当然，您的老板也出席了这个会议。您感到非常惭愧，恨不得找个地缝钻进去。收费昂贵的专家到了会场并进行数据库分析，而此时您坐在一个不起眼的位置上。这位专家给出了一份带有图表和曲线的多页、彩色报告，令您的老板和用户眼花缭乱，之后便离开了。无论专家推荐的方案是否有效，您的名声已经受损。您的调整方法令您和老板失望，您变成了一位“调整恐惧症患者”。

那时候比率数是仅有的交流形式。我们不得不使用一些数值，尤其是用缓冲存储器命中率来表示性能，这并不容易。我们总是疑惑，当告诉用户数据库高速缓存命中率很高而完全不顾他们对响应时间很慢的抱怨时，为什么他们的眼神显得不解和迷茫。很明显，我们的语言对他们来说就像外星人的语言，但我们不能理解为什么他们无法理解。我的意思是说，它有多难理解吗？高速缓存命中率高是好事，不应有问题；高速缓存命中率低才不好，才会出现问题，不是吗？但是比率数对那些想知道为什么他们的任务运行缓慢的人来说实在毫无意义。想像一下如果您打电话了解为什么比萨饼迟迟没有送到，人家却告诉您司机的命中率是 65%，那您肯定会觉得莫名其妙。又或者假设您的老板问您为什么上班迟到。您可能将迟到归因于一个或更多的事件，比如一次交通事故，变化莫测的驾驶条件、缓慢的交通、交通灯等。您不可能向老板解释您的迟到是因为某种比率。

高速缓存命中率方法对性能分析和调整完全不可靠，否则 Oracle 公司也不会引入新的方法。高速缓存命中率高表示性能好这一说法往往是错误的。现实中，高命中率并不总意味着良好性能，而低命中率也不总意味着不良性能。实际上，很多时候高速缓存命中率一降到底，但是应用程序性能并未降低。试图通过高速缓存命中率数值判断性能就像试图用转速表读数来确定车速一样。没有速度计，您无法告诉别人每分钟转速为 6000 时车到底有多快。转速表读数很高，但是汽车可能挂在二档。

Oracle Database 10g Release 1 Performance Tuning Guide 一书中作了如下阐述：