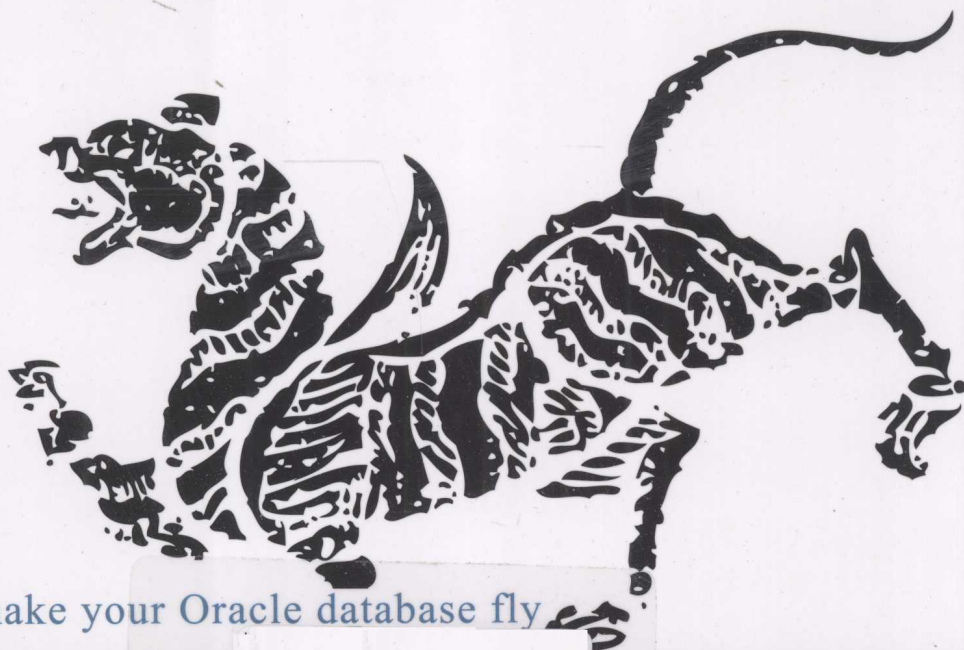


国内第一本真正意义上从工作经验出发，  
以作者的心得体会全面论述Oracle数据库性能优化的书籍。



How to make your Oracle database fly

# 让Oracle跑得更快

## Oracle 10g 性能分析与优化思路

谭怀远 著



电子工业出版社  
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY  
<http://www.phei.com.cn>

IT **PUB** 技术丛书



# 让Oracle跑得更快

## Oracle 10g 性能分析与优化思路

谭怀远 著

电子工业出版社  
Publishing House of Electronics Industry  
北京•BEIJING

## 内 容 简 介

在这本书里读者将会学到作者在性能优化方面的一些思路和思考,一些故障处理的方法和原则,这些东西是作者在实践中长期积累的心得体会,当读者掌握了一些处理问题的基本思路之后,成为一名合格的DBA 就是一件轻而易举的事情了。

本书适用对象: Oracle DBA、Oracle 开发人员,和其他对 Oracle 数据库感兴趣的人员。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。  
版权所有,侵权必究。

## 图书在版编目(CIP)数据

让 Oracle 跑得更快: Oracle 10g 性能分析与优化思路 / 谭怀远著. —北京: 电子工业出版社, 2010.8  
(ITPUB 技术丛书)  
ISBN 978-7-121-11325-3

I. ①让… II. ①谭… III. ①关系数据库—数据库管理系统, Oracle 10g IV. ①TP311.138

中国版本图书馆 CIP 数据核字 (2010) 第 131770 号

责任编辑: 高洪霞

印 刷: 北京市天竺颖华印刷厂

装 订: 三河市鑫金马印装有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编: 100036

开 本: 787×980 1/16 印张: 28.5 字数: 533 千字

印 次: 2010 年 8 月第 1 次印刷

印 数: 4000 册 定价: 59.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 [zlts@phei.com.cn](mailto:zlts@phei.com.cn), 盗版侵权举报请发邮件到 [dbqq@phei.com.cn](mailto:dbqq@phei.com.cn)。

服务热线: (010) 88258888。

# 序

性能优化是数据库应用的核心问题。目前的商业或开源的数据库产品，发展已日臻成熟，很少有经常发生崩溃急需修复的情形。故 DBA 除了日常的常规维护任务外，大多把精力花在优化数据库上。在 2005 年的时候，ITPUB 也曾出过一本《Oracle 数据库性能优化》（盖国强、冯春培、叶梁、冯大辉主编）的技术书，颇受 Oracle DBA 们的欢迎。现在很高兴地看到在 ITPUB 技术丛书里又增加了一本关于数据库优化知识的书籍。

ITPUB 于 2001 年 9 月 26 日成立的，已发展为国内最大的数据库技术讨论社区。本书作者谭怀远先生则是在 2001 年 9 月 28 日加入了 ITPUB，相交至今将近九年时间。他在这么长久的时间，长期工作在数据库业务的第一线，积累了大量丰富的经验，也形成自己独到的见解。而这些见解，又大部分体现在本书的文字里，本书既是知识的归纳总结，同时又是个人技术感情（怨我在这里使用了一个创新的词汇，大多数长期从事技术工作的人，都有一种有墨在胸，不得不发的感觉）的抒发。纵观全书，我个人对作者所说的“最难的东西不是技术本身，而是什么时候该用什么技术”深表认同。当 DBA 从烦琐的日常工作脱身出来，举目远望的时候，再往前的一片田野便是架构问题，最好的最彻底的，能一劳永逸的优化，往往从架构设计开始。期待怀远君将来的新作，可以在这片更广阔的天地里驰骋。

我感到本书最大的一个特点在于，作者通过自己的一种情绪化的东西在写作，也可以说是对于技术的一种感情告白，所以是真挚的，这种真挚将影响到读者阅读时的情绪，让你在一种颇为感性和人性化的氛围里阅读，轻松而又有趣，而不是冷冰冰的枯燥的技术讨论，这是本书区别于其他技术类书籍的一个显著特点。

本书里，涵盖了几乎所有优化相关的知识点，以及一些很新的内容，比如 bind peeking、并行执行、执行计划、Cardinality（基数）、10053 事件等，这些内容对于那些渴望深度了解性能优化的读者来说，是非常有用的。

ITPUB 前身是在 smiling 上的 Oracle 电子小组，刚刚开始的时候 ITPUB 的板块不多，只有 Oracle 数据库管理、Oracle 开发、OCP、Cisco、网络集成、海阔天空这几个板块，会员数只有一万多人，今年过 9 年的发展，ITPUB 已拥有技术板块 100 多个，注册会员数量超过 230 万人。每天更新的讨论帖数以万计。ITPUB 的发展与像怀远君这样的专家、第一线技术工作者的长期支持是分不开的，在此也向怀远君表示感谢，没有你们就不会有今天的 ITPUB。

ITPUB 创始人 tigerfish

2010 年 6 月 8 日

很多年前就在 itpub 看到作者的身影，也了解到作者管理着大量的数据库。多年的不断锤炼让作者在 Oracle 数据库领域有了丰富的经验。在数据库优化领域，国内的书籍相对比较少，不论哪种数据库，比较多的都是类似工具手册一样。写书是一件很不容易的事情，对作者的知识体系有着极高的要求，所以市面上流传的很多都是简单地将英文的文档翻译为中文的手册类的书籍。作者将自己多年的经验用自己的语言和通俗的比喻给我们展示出来，带给人的是另一种体验，更亲切和容易理解。除了常规的优化所涉及的范畴及 Oracle 10g 开始推出的 AWR 和 ASH 之外，作者还引出了 10053 这样的 CBO 相关的事件及不少的 Hint 方法，这些都将帮助我们非常深入地研究数据库的 SQL 优化问题。相信作者将亲身经历的体验深入浅出地展示给我们，能给 Oracle 数据库爱好者很好的帮助。

——冯春培 (biti)

当我们在 2004 年开始编辑出版 Oracle 技术书籍时，国内原创的作品还十分有限，现在，这种情况完全改变了，越来越多的技术爱好者开始总结、写作和分享，Oracle 技术出版物开始丰富起来。在这个历程中，ITPUB 论坛一直推动着 Oracle 数据库技术的探讨和应用，作者 alantany 正是来自于 ITPUB 的一位技术专家，他将自己多年的实践与经验不断总结出来，和我们大家分享，这种精神与坚持值得我们尊敬，我乐于见到这样的作品问世，也期待作者能够坚持不懈，不断同我们分享他的知识与经验。

——盖国强 (eygle) 恩墨科技创始人，Oracle ACE 总监

很欣喜地看到这几年来国内 Oracle 数据库技术原创书籍的蓬勃发展，对每一位能够坐下来并且将自己的经验写出来的作者我都报以深深的敬意。对于 Oracle 数据库而言，也许在现在随着安装的日渐简易和默认性能的不断优化，一份默认的数据库安装，甚至是保证数据库可以正常运行都不再需要太多的专业知识，然而数据库优化仍然是非常专业的部分，这需要多年的实际工作经验积累。可以说这本书的作者拥有得天独厚的工作环境，据我所知，他所参与管理的数据库无论是数量还是大小还是性能要求上在国内都可以排入前列，因此这是一本融合了真知灼见、可以指导实际工作的 Oracle 数据库性能优化书籍。实际上我更希望读者们可以从后记开始读起，作者在后记中提及的所有观点都与我不谋而合，我同样相信无论是后记中这些观点还是全书中记录的技术知识，对于所有从业人员都有极大帮助。

——业内资深 Oracle 技术专家，Oracle ACE 张乐奕 (Kamus)

据我所知作者是中国独立撰写 Oracle 性能优化书籍的第一人，该书几乎涵概了 Oracle 性能优化的所有主题，在国内数据库性能优化领域的书籍中实属罕见。

——谢永生 (warehouse) 资深 Oracle 培训讲师

The fast and easy way to understanding the fundamentals of database performance tuning. If you're tired of wading through huge technical manuals that drown you in jargon, making it difficult to decipher database performance issues, help has finally arrived. Simple enough for a beginner, but challenging enough for an advanced user, this book is your shortcut to mastering database performance tuning.

本书提供了一种理解 Oracle 性能优化的简单快捷的方法。如果你已经被成堆的关于性能优化的技术手册淹没，身心疲惫，无从下手，那么，这本书将帮你走出困境。无论你是初学者，或是经验丰富的技术人员，你都将在这本书中获得一个掌握数据库性能优化的捷径。

——Mike ITG (Investment technology group) 资深软件工程师

# 前言

笔者在写这本书的时候，翻看了很多当前国内数据库方面的书籍，发现写性能优化的书并不多，特别是从工作经验和思路上来讨论性能方面的书，更是少之又少，这些因素让笔者思考要写这样一本书，这也算是这本书的一个定位。

在这本书里，你将会学到笔者在性能优化方面的一些思路和思考，一些故障处理的方法和原则，这些东西是笔者在实践中长期积累的心得体会，在笔者看来，掌握处理问题的方法和分析问题的思路在日常工作中显得更为重要，当你掌握了一些处理问题的基本思路之后，剩下的工作就是去 Google 或者阅读参考书了。

本书的一个特点是，凡是作者提到的观点，都尽可能地使用一些例子来证明它，这样看起来更有说服力一些。

## 为什么会出现数据库的性能问题

性能问题是最近几年来 DBA 们越来越关注的一个数据库技术领域，归根结底，造成它的原因是最近几年信息化进程的飞速发展，导致了很多系统的用户数量猛增，数据库中存储的数据量亦成几何级数激增，数据库作为数据处理的最终受体，将必然直接承担这种变化导致的性能下降。因此在人们对信息的依赖性越来越强的时候，对信息使用的效率也变得越来越关注，这样数据库的性能优化问题就日益严重地压在 DBA 的身上。

## 什么时候需要对性能进行干预

itpub.net 是国内最早的一个专业讨论 Oracle 数据库技术的论坛，目前在国内数据库方面已经相当有知名度，笔者是 2001 年注册的，算是最早的会员之一。目前仍然会经常

上去看看，由于工作内容的关系，我比较关注性能方面的帖子，发现以下一类的帖子经常有很多，比如：

1. 我是一个 DBA，我现在手头有一个数据库，我该从哪里进行性能优化呢？
2. 这是我的数据库的一个 Statspack，我该如何优化？

通常对于第一个问题，我是很少回答的，并不是不屑于回答，实在是没有办法回答，如果我回答说，“你怎么知道你的数据库需要优化？”又担心这种没有实际意义的回答带有说教意味，打击别人的积极性，所以通常看看而已。实际上我是想说，对于一个 DBA 来讲，当你拿到一个数据库的时候，你首先需要做的是用最短的时间来了解一下跑在这个库上的是一个什么系统，比如是在线事务（OLTP）系统还是在线分析（OLAP）系统，这对于你做出性能上的判断至关重要，如果连系统都不了解，真不知道该如何去优化它，这就好比说，要设计一辆汽车，如果连用户对汽车的喜好都弄不清楚，如何能设计出一个取悦于用户的车呢？

对于第二个问题，像是比第一个具体一些，因为帖子作者已经提供了一个性能数据报告，但我仍然觉得通过这些数据没有办法准确地判断数据库是否有性能问题。比如你是一个医生，我让一个人站在你的面前测心率，结果是 50 次/分钟，你是不是可以断定他有问题，需要安装心脏起搏器呢？实际上是不需要，因为我知道他是一个运动员，这样的心率是正常的，而医生不知道，所以他在做出诊断之前需要详细了解站在自己面前的应诊者的所有详细信息，来作为他做出判断的依据。

下面贴出一个来自于我使用过的数据库性能报告中的一部分：

|                               |        |                   |        |
|-------------------------------|--------|-------------------|--------|
| Buffer Nowait %:              | 100.00 | Redo NoWait %:    | 99.99  |
| Buffer Hit %:                 | 66.35  | In-memory Sort %: | 100.00 |
| Library Hit %:                | 99.63  | Soft Parse %:     | 96.87  |
| Execute to Parse %:           | 83.99  | Latch Hit %:      | 99.87  |
| Parse CPU to Parse Elapsed %: |        | % Non-Parse CPU:  | 100.00 |

许多人看到这个数据一定会大声说：

“嘿，你的数据库性能好差，buffer hit 只有 66%，不知道是谁设计的这个系统，赶

紧加大 data buffer 的尺寸!”

诚然，这个数据的确显示数据库的内存命中率低得可怜，但是我想告诉你的是，这是一个在线分析（OLAP）系统的数据库，运行着很多非常大的查询，每个查询搜索的范围都在上亿条记录以上，那这个结果不是很正常吗？我们需要把几亿条数据缓存到内存里提供给这种每天可能只运行几次的查询吗？你可以同意，但是你的老板是不会同意的，这样做的成本太高了，而且完全没有必要，因为它只是一个报表系统，对数据库的响应时间要求不高，所以我们当然可以让这个查询直接到磁盘上去搜索数据，这也就是为什么在这样的系统里，buffer hit 比例很低，但却是一个完全可以接受的值的原因。

笔者认为，只有数据库的性能已经影响到业务的正常工作或者用户已经无法满意于这种性能时，我们才应该考虑来优化它，对于绝大多数系统，数据库的安全和稳定才是最重要的。

### FAST=TRUE?

这是很多人追求的目标，它的意思是，在 Oracle 数据库中，通过调整性能参数的值，就可以让数据库运转得飞快。

实际上这不过是句玩笑，它本身是一句反话，却让很多人误入歧途。我看到很多人，包括一些 DBA，凡涉及性能优化，必定谈及性能参数的修改，这实在是一个误区，他把性能参数值的修改对数据库性能的正面影响人为地放大了很多倍，实际上恰恰相反，很多时候修改这些参数产生的却是副作用，原因很简单，Oracle 给一个参数一个默认值是让它最大限度地适用于每个数据库，所以它几乎是最优的，当然，绝对有个别数据库需要适当调整，但我认为那是个例，并且，很多时候，修改这些参数的人，他们修改的理由并不是非常充分，不过是想修改一下看看运气而已。

## 本书的内容



以下是本书各个章节的内容简介。

### 第 1 章 引起数据库性能问题的因素 ■ ■ ■

这一章主要讨论一些引起数据库性能问题的因素，包含了系统架构、软件代码、数据库设计、存储设计等话题。

### 第 2 章 锁和阻塞 ■ ■ ■

在这一章里，将介绍 Oracle 数据库中锁的起因及由锁引起的性能问题——阻塞，并

将讨论常见的几种阻塞的起因。

### 第 3 章 Latch 和等待 ■ ■ ■

这一章讨论 Latch，它是 Oracle 中比锁更轻量级的一种串行机制。热块或是 SQL 未绑定变量是最常见的导致 Latch 等待的原因，这一章将对这些成因及解决方法进行论述。

### 第 4 章 优化器 ■ ■ ■

优化器是 SQL 执行中最核心的部分，如果要分析 SQL 的性能，就不能不了解 Oracle 优化器的机制，这一章，我们就带你走进 Oracle 优化器——CBO 的世界。

### 第 5 章 执行计划 ■ ■ ■

当我们分析一条 SQL 语句的性能时，最先做的事情大概就是分析它的执行计划了。

所以，如果连执行计划都看不懂，那 SQL 调优根本无从谈起。在这一章，我们将讨论 CBO（基于成本的优化器）执行计划相关的内容。

### 第 6 章 Hint ■ ■ ■

Hint 指通过人为的方式来约束 SQL 的执行计划，让它按照我们希望的方式来执行，以达到我们需要的目的——改善性能或者仅仅是试验以对比 SQL 的执行性能。

这一章将讨论 Oracle 数据库中的大多数 Hint。

### 第 7 章 分析及动态采样 ■ ■ ■

对象采样分析是 CBO（基于成本的优化器）的灵魂和核心，CBO 如果没有了对象的分析数据，就好像一个医生不使用病人的病历来确定病人的病一样危险——那是一种没有依据的、盲目的行为。

在这一章里，我们将详细讨论 Oracle 中和对象分析相关的内容。

### 第 8 章 并行执行 ■ ■ ■

这一章讨论一个和性能关系极大的技术——并行执行。

在 OLAP（在线分析系统）或者是数据仓库系统中，并行技术使用得非常普遍，在合适的条件下，并行执行将会使 SQL 的执行效率大幅度提升。

## 第 9 章 变量绑定 ■ ■ ■

这一章将详细讨论一个在性能优化领域经常被谈到的话题——变量绑定。

那么，是不是在任何时候变量绑定都是必需的呢？答案是否定的，在这一章中将给出答案。

## 第 10 章 SQL\_TRACE 和 10046 事件 ■ ■ ■

SQL\_TRACE 和 10046 事件是会话级非常有用的两个工具，它们可以捕获会话当中 SQL 执行的详细信息，其中 10046 事件还可以获得 SQL 绑定变量的信息及发生的等待事件。

这一章将详细讨论这两个工具。

## 第 11 章 10053 事件 ■ ■ ■

这一章将详细讨论 10053 事件，它是一个很有用处的工具，当你发现一条 SQL 总是选择错误的执行计划，而你又百思不得其解的时候，也许你应该去生成一个 10053 事件的 trace 文件，看看 CBO 究竟是如何做出这样的执行计划的。

## 第 12 章 性能视图和性能参数 ■ ■ ■

本章讨论一些 Oracle 数据库的性能视图和性能参数。

性能视图相对于 SQL\_TRACE 来说，可以让我们更直接地获取一些性能数据，帮助我们判断数据库是否出现了性能问题。

而性能参数则让我们能够有机会选择一种最适合自己的系统的某个参数值，以最大程度地满足当前系统的需要。

## 第 13 章 性能报告 ■ ■ ■

本章介绍了常用的几个性能分析工具及性能报告，包括 AWR, STATSPACK 和 ASH，其中以 AWR 性能报告作为重点介绍的对象。

本章以一个来自于现实生产数据库的 AWR 报告为题材，来讨论 AWR 报告的阅读方式，并最终判断出系统的性能所在；STATSPACK 介绍了它的安装方法和如何生成报告；ASH 也是以一个来自实际生产数据库的性能报告进行性能分析。

## 附录 A 常见的等待事件 ■ ■ ■

这部分会列出一些常见的等待事件、引起它们的原因及一些内部的机制，可以作为大家在处理性能问题时的一个参考部分。

## 后记 关于数据库的学习方法 ■ ■ ■

这一部分是作者对如何学习 Oracle 的一个心得分享，对于 Oracle 初学者来说，正确的学习方法非常重要，它可以使你少走很多弯路。

如果初学者对于 Oracle 数据库的学习方法有兴趣，这部分可以作为本书的第一部分来阅读。

## 本书的读者对象

■ -----

1. 本书适合 Oracle DBA 或者和 Oracle 相关的开发人员。
2. 本书的读者需要有一定的 Oracle 基础，比如你应该知道什么叫做表，什么叫做索引等。

## 约定 ■ ■ ■

1. 本书示例使用的 Oracle 版本是 10gr2。
2. 本书自创了一个术语——段对象。

这个词大家可能看着有那么一点陌生。作者的意思是，在 Oracle 数据库中，凡是分配了存储空间的，都称为段，所以段并不一定指的是表，也可能是表的一个分区，还可能是索引、大对象（LOB），或是 IOT（索引表），物化视图等。在书中有时候需要描述这些对象时，单独说某个表，或者一个索引，都不能完全概括，所以就统称为段对象。

3. 几个未作翻译的术语 Extent, Latch 和 Bind peeking。

**Extent:** 我看有些书翻译为“分区”，说实话，在 Oracle 里面，一提到分区，可能 99% 以上的人会认为是 partition，还有的书翻译为“范围”，这个就更让人匪夷所思，所以在书中这个单词就没有翻译，相信大家也懂。

**Latch:** 有的书翻译成“闩”，有的翻译成“锁存”，我总觉得还是不翻译好，只要

大家知道它是 Oracle 里一种类似于锁的保证一些操作串行化的技术就好了。

**Bind peeking:** 翻译成“变量窥视”或是“变量窥探”都非常不对头，所以干脆也不翻译。

## 本书的目的 ■ ■ ■

笔者从事 Oracle DBA 的工作已经超过 10 年，对数据库的理解也一直在改变，就目前来看，我觉得最难的东西不是技术本身，而是什么时候该用什么技术。比如说要使用变量绑定，这非常容易，如果你不会，Google 一下，差不多几分钟时间你就会了。可是，这个系统究竟该不该使用变量绑定，我想你 Google 一天或者一个星期也不一定有答案。原因是每个系统都是独立的，都有自己的业务特点，这需要技术人员根据自己系统的业务特点来度身定做符合自己系统的技术特性。

让读者在每一个技术面前先停下来思考一下，这个技术究竟在什么时候应该用，什么时候不应该用，这是笔者写本书的最终目的。

# 致 谢

感谢 [www.itpub.net](http://www.itpub.net) 网站的创始人 Tigerfish 为本书写序，我一直对他怀有敬意，他在推动中国 Oracle 数据库的发展上功不可没。

感谢 biti（冯春培），eygle（盖国强），kamus（张乐奕）和 warehouse（谢永生），他们都是国内顶尖的 Oracle 专家，感谢他们为本书写的精彩点评。

还有来自 ITG 的 Mike，感谢他的热心帮助和鼓励。

感谢 ITPUB 的王蓓小姐（贝贝），在本书的出版过程中，她做了大量的协调工作，才保证这本书的顺利出版。

感谢电子工业出版社的张月萍策划和高洪霞编辑，是她们的努力让本书更具可读性和完整性。

最后要感谢我的妻子 tracy 和儿子思墨，是他们让我一直努力工作，最终使本书得以问世。

# 目 录

|                               |    |
|-------------------------------|----|
| 第 1 章 引起数据库性能问题的因素 .....      | 1  |
| 1.1 软件设计对数据库的影响 .....         | 1  |
| 1.1.1 软件架构设计对数据库性能的影响 .....   | 1  |
| 1.1.2 软件代码的编写对数据库性能的影响 .....  | 2  |
| 1.2 数据库的设计 .....              | 8  |
| 1.2.1 OLTP 数据库 .....          | 9  |
| 1.2.2 OLAP 数据库 .....          | 10 |
| 1.3 数据库的硬件设计 .....            | 14 |
| 1.3.1 存储容量 .....              | 15 |
| 1.3.2 存储的物理设计 .....           | 16 |
| 1.3.3 数据的安全 .....             | 17 |
| 1.4 小结 .....                  | 19 |
| 第 2 章 锁和阻塞 .....              | 20 |
| 2.1 关于锁 .....                 | 20 |
| 2.2 锁和阻塞 .....                | 22 |
| 2.3 引起阻塞的其他情况 .....           | 30 |
| 2.3.1 select for update ..... | 30 |
| 2.3.2 外键和索引 .....             | 36 |
| 第 3 章 Latch 和等待 .....         | 44 |
| 3.1 共享池中的 Latch 争用 .....      | 45 |
| 3.2 数据缓冲池 Latch 争用 .....      | 54 |
| 3.2.1 表数据块 .....              | 54 |
| 3.2.2 索引数据块 .....             | 59 |
| 3.2.3 索引根数据块 .....            | 62 |
| 3.2.4 段头数据块 .....             | 65 |

|                                               |     |
|-----------------------------------------------|-----|
| 第 4 章 优化器 .....                               | 66  |
| 4.1 RBO 基于规则的优化器 .....                        | 66  |
| 4.2 CBO 基于成本的优化器 .....                        | 69  |
| 第 5 章 执行计划 .....                              | 85  |
| 5.1 Cardinality (基数) .....                    | 85  |
| 5.2 SQL 的执行计划 .....                           | 94  |
| 第 6 章 Hint .....                              | 109 |
| 6.1 和优化器相关的 Hint .....                        | 115 |
| 6.1.1 all_rows 和 first_rows (CBO) .....       | 115 |
| 6.1.2 RULE Hint .....                         | 117 |
| 6.2 访问路径相关的 Hint .....                        | 117 |
| 6.2.1 FULL Hint .....                         | 118 |
| 6.2.2 INDEX Hint .....                        | 118 |
| 6.2.3 NO_INDEX Hint .....                     | 118 |
| 6.2.4 INDEX_DESC Hint .....                   | 119 |
| 6.2.5 INDEX_COMBINE Hint .....                | 119 |
| 6.2.6 INDEX_FFS .....                         | 119 |
| 6.2.7 INDEX_JOIN .....                        | 120 |
| 6.2.8 INDEX_SS Hint .....                     | 120 |
| 6.3 表关联顺序的 Hint .....                         | 125 |
| 6.3.1 LEADING Hint .....                      | 125 |
| 6.3.2 ORDERED Hint .....                      | 126 |
| 6.4 表关联操作的 Hint .....                         | 127 |
| 6.4.1 USE_HASH, USE_NL 和 USE_MERGE Hint ..... | 127 |
| 6.4.2 NO_USE_HASH Hint .....                  | 132 |
| 6.4.3 NO_USE_MERGE Hint .....                 | 133 |
| 6.4.4 NO_USE_NL Hint .....                    | 133 |
| 6.5 并行执行相关的 Hint .....                        | 134 |
| 6.5.1 PARALLEL Hint .....                     | 134 |
| 6.5.2 NO_PARALLEL Hint .....                  | 134 |
| 6.6 其他方面的一些 Hint .....                        | 135 |
| 6.6.1 APPEND Hint .....                       | 135 |
| 6.6.2 DYNAMIC_SAMPLING Hint .....             | 135 |
| 6.6.3 DRIVING_SITE Hint .....                 | 136 |
| 6.6.4 CACHE Hint .....                        | 136 |
| 6.7 小结 .....                                  | 136 |

|                                           |     |
|-------------------------------------------|-----|
| <b>第 7 章 分析及动态采样</b> .....                | 138 |
| 7.1 直方图.....                              | 141 |
| 7.2 DBMS_STATS 包.....                     | 147 |
| 7.3 动态采样.....                             | 176 |
| 7.3.1 什么是动态采样.....                        | 176 |
| 7.3.2 动态采样的级别.....                        | 182 |
| 7.3.3 什么时候使用动态采样?.....                    | 185 |
| 7.4 小结.....                               | 185 |
| <b>第 8 章 并行执行</b> .....                   | 186 |
| 8.1 并行和 OLAP 系统.....                      | 187 |
| 8.2 并行处理的机制.....                          | 189 |
| 8.3 读懂一个并行处理的执行计划.....                    | 191 |
| 8.4 一个很常见的并行执行等待事件.....                   | 192 |
| 8.5 并行执行的适用范围.....                        | 194 |
| 8.5.1 并行查询.....                           | 194 |
| 8.5.2 并行 DDL 操作.....                      | 195 |
| 8.5.3 并行 DML 操作.....                      | 203 |
| 8.6 并行执行的设定.....                          | 210 |
| 8.6.1 并行相关的初始化参数.....                     | 210 |
| 8.6.2 并行度的设定.....                         | 211 |
| 8.7 直接加载.....                             | 213 |
| 8.7.1 直接加载和 REDO.....                     | 216 |
| 8.7.2 直接加载和索引.....                        | 219 |
| 8.7.3 直接加载和并行.....                        | 221 |
| 8.7.4 直接加载和 SQL*LOADER.....               | 226 |
| <b>第 9 章 变量绑定</b> .....                   | 232 |
| 9.1 什么是变量绑定, 为什么要做变量绑定.....               | 232 |
| 9.2 为什么说 OLTP 必须要求变量绑定而 OLAP 不应该绑定变量..... | 241 |
| 9.3 bind peaking.....                     | 248 |
| <b>第 10 章 SQL_TRACE 和 10046 事件</b> .....  | 254 |
| 10.1 SQL_TRACE.....                       | 254 |
| 10.2 TKPROF 工具.....                       | 256 |
| 10.3 10046 事件.....                        | 268 |

|        |                                         |     |
|--------|-----------------------------------------|-----|
| 第 11 章 | 10053 事件 .....                          | 276 |
| 第 12 章 | 性能视图和性能参数 .....                         | 294 |
| 12.1   | 性能视图 .....                              | 294 |
| 12.1.1 | V\$SQL .....                            | 295 |
| 12.1.2 | V\$SQL_SHARED_CURSOR .....              | 300 |
| 12.1.3 | v\$session .....                        | 305 |
| 12.1.4 | V\$sesstat .....                        | 309 |
| 12.1.5 | V\$session_wait .....                   | 310 |
| 12.2   | 性能参数 .....                              | 312 |
| 12.2.1 | Cursor_sharing .....                    | 313 |
| 12.2.2 | DB_FILE_MULTIBLOCK_READ_COUNT .....     | 328 |
| 12.2.3 | PGA_AGGREGATE_TARGET 和 SGA_TARGET ..... | 334 |
| 12.2.4 | OPTIMIZER_DYNAMIC_SAMPLING .....        | 334 |
| 第 13 章 | 性能报告 .....                              | 335 |
| 13.1   | AWR 性能报告 .....                          | 335 |
| 13.1.1 | 生成 AWR 性能报告 .....                       | 337 |
| 13.1.2 | AWR 性能报告分析 .....                        | 342 |
| 13.2   | Statspack 性能报告 .....                    | 386 |
| 13.2.1 | Statspack 的安装 .....                     | 386 |
| 13.2.2 | Statspack 性能采集 .....                    | 391 |
| 13.3   | ASH 性能报告 .....                          | 394 |
| 13.3.1 | 生成 ASH 性能报告 .....                       | 395 |
| 13.3.2 | ASH 性能报告分析 .....                        | 405 |
| 13.4   | 小结 .....                                | 416 |
| 附录 A   | 常见的等待事件 .....                           | 417 |
| 后记     | 关于数据库的学习方法 .....                        | 434 |