

<<DBA的思想天空>>

图书基本信息

书名：<<DBA的思想天空>>

13位ISBN编号：9787115294432

10位ISBN编号：7115294437

出版时间：2012-10-18

出版时间：人民邮电出版社

作者：白 鱗,储学荣

页数：432

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

## &lt;&lt;DBA的思想天空&gt;&gt;

## 前言

写完《Oracle优化日记：一个金牌DBA的故事》和《ORACLE RAC日记》后，很多网友问我下面是否会继续写书。

我的想法还是和以前一样，先整理整理自己的思路，写一些东西发在Oracle粉丝网上，写过一段时间再根据已经完成的内容决定新书的结构。

开始我是想把新书写成DBA日记系列第三部的，不过写作的过程中才发现这种模式写下去有一点千篇一律的感觉了，很多案例从根本上看十分相似。

有一次和同事老储聊天的时候，他提到现在很多年轻人不会按照Oracle的内在原理去考虑问题，从而导致经常出现常识性的错误。

他的这句话就像火花一样在我脑海中闪现，Thinking in Oracle这几个英文词汇就出现在我的脑中。

如果每个人都能以Oracle的基本原理为依据去考虑问题，那不是很好吗？

老储的英文名字是John，在我的朋友里有两个John。

一个是老外John，我还在玩ICQ的时候认识的网友，一个Oracle技术狂人。

不过老外John现在已经成为一家银行的IT技术主管，随着岁月的流逝，当年的技术狂人现在已经成了狂热的人文主义者。

去年圣诞新年假期他刚刚完成了10000公里行程的中南美洲自驾游，所到之处全部入住当地最高级的酒店，吃当地最昂贵的美食，还常常邂逅美女，虽然邮件还只是寥寥数字，但是羡慕嫉妒恨已经让我把这个资本主义的老家伙好好骂了几天。

另外一个John就是老储，他属于睡在我对面的弟兄，我大学时的室友。

自从1995年我把他从unixware汉化小组忽悠到深圳后，我们一直在一起合作。

老储是个很低调的人，低调到第一次和客户打交道的时候客户可能会质疑他的能力，不过随着和他交往的深入，你可以发现他的深不可测。

老储的建议让我重新定义了本书的结构，把它分为基础知识、工具、方法和案例四大部分。

不过随着写作的深入，我发现这个工作是十分艰巨的。

这样一本书结构之大，内容之庞杂，远远超出了我的想象。

于是我重新调整结构，将工具这一部分的内容从书中拿掉，准备独立成书，把方法和案例用两章讲完。

这样调整后本书的内容更为紧凑，篇幅也可以控制在450页左右。

更大的惊喜也接踵而来，我终于说服了老储，让他参与到本书的写作中来。

其实第一次和老储沟通这本书的时候我就邀请过他，不过被他以没空为由拒绝了。

老储是一个懒散的人，我每次给他布置写文档的任务，他总是在快到期的时候才开始动手，不过每次提交的东西都让人无可挑剔。

能让这么靠谱的人参与本书的写作，确实可以为书增色不少。

老储凭着深厚的开发功力，在性能优化方面具有很敏锐的观察力，往往能够在很短很短的时间内找到系统的关键问题，而且他在Oracle文件结构和ASM的结构方面的研究很深，自己也编写了一个类似DUL的工具。

有了他的加入，关于ASM原理和数据文件结构这部分内容我就可以推给他了。

此前我正为这两节犯愁，考虑是否在书中去掉这两节，补充一些其他内容。

老储的加入使这两节保留了下来，对于数据文件结构和ASM文件结构感兴趣的朋友可能会感到庆幸，保留这两节也使得本书的内容更加丰富。

我给本书起的名字是《Thinking in ORACLE之DBA的思想天空》，主编觉得这是一个十分霸气的名字。

实际上，透过某个事物的本质去看问题，无论针对什么，都是比较高的境界。

对于某些事情，在没有弄清楚其本质之前，我们往往难以找到正确的应对方法，虽然偶尔我们会像瞎猫碰到死耗子一样歪打正着，但是好运气不会总是伴随着你。

就像前些年，我不了解“回南天”的成因，因此在每年的2、3月份总是十分痛苦。

## &lt;&lt;DBA的思想天空&gt;&gt;

在广东沿海生活过的人都知道每年的春天总是会碰到回南天，时间有长有短，至少也在半个月左右。在回南天里，家里到处都是湿漉漉的，地上、墙上甚至天花板上都会渗出水珠。

在这样的环境中生活半个月，绝对是十分恐怖的事情。

为了让家里尽快干起来，我的第一反应是门窗大开，同时用风扇拼命扇风。

不过这样处理并不能减轻返水的现象，有时候反而水更加多了。

后来我请教了一位搞大气海洋研究的人，他告诉我回南天的成因是春天东南季风带来大量的水气，而当气温回升的时候，室外气温高于室内气温，湿热的空气遇到室内较冷的物体时，就会发生冷凝现象，从而就引发了反水的现象。

一旦了解了回南天的成因，就很容易找到对付回南天的办法了，只要碰到气温大幅度回升的天气，就门窗紧闭。

靠着这个办法，我终于摆脱了回南天的困扰，无论门外的走廊湿成什么样，我家的地上总是干干的。

后来每当我看到朋友家里满地积水的时候，就会把这个方法教给他们，他们也逐渐远离了回南天的困扰。

从这个生活案例中，我们也可以看到，一旦了解了问题的本质，就很容易找到正确的解决方法。

而没有理解问题本质的时候，我们所采取的应对措施不一定是靠谱的。

这个原则应用到Oracle数据库方面，也是一样的。

对于每个来应聘DBA的人我都会问他们一个问题：“Oracle到底是什么？”

“有些人会用数据库基础的理论来回答我：“数据库是数据的集合。”

“也有些人会感到茫然，不知道我问这个问题是什么意思。

实际上很多Oracle DBA从来没有思考过这个问题。

“Oracle就是Oracle，是一个产品，还能有什么意思呢？”

我不知道Oracle到底是什么也没有影响到我做一个合格的DBA。

“很多人都会这么想。

实际上对于Oracle我们确实还需要重新去认识认识，每个DBA在学习Oracle的时候都往往注重于学习如何建库、如何管理、如何编程、如何优化。

我们总在不停地去学习一些方法，学习一些秘籍。

如果偶尔学到了一些不传之秘，都会感到兴奋异常。

也有些人使用这些秘籍解决了一些疑难杂症，成为了大家传说中的高手。

虽然说这也是学习Oracle数据库最为常见的一种方法，但是这样学习下去，我们总是在记忆一些枯燥的语法和脚本，虽然经过数年我们积累下了大量的经验，但还是无法真正地理解Oracle，数据库升级了，系统变化了，我们就必须从头去学习。

常年累月，我们总是在一次又一次循环往复地重复着同样的事情，直到筋疲力尽，对Oracle失去往日的激情，最终DBA只是一个职业，Oracle只是我们谋生的手段。

这样学习下去，几年后，很多人就会碰到瓶颈，虽然说自己处理问题的能力和工作经验已经很丰富了，但是技术好像停滞不前了。

我周围一些做了五六年DBA工作的朋友都遇到过类似的情况，他们咨询我的时候，我告诉他们，这是因为经验的积累已经到了一定程度，需要对Oracle基础概念有更深刻的认识。

这种情况下，你需要静下心来认真看书，学习Oracle的基础概念，只有彻底搞清楚了这些，才能跨过这道坎，达到一个新的境界。

绝大多数工作了五六年的朋友已经无法静下心来做这些事情了，因此他们失去了突破的机会。

不过也没关系，大多数人选择了新的职业规划，从事管理，或者转向售前、业务专家等职位。

事实上，我们可以换一种方式来学习Oracle，让Oracle的精神融入DBA的血液中，让DBA像Oracle一样思考问题，使Oracle成为我们的爱好，作为我们生命的一部分存在。

对于大多数DBA来说，这也许只是一个乌托邦式的理想，多数DBA只是需要有一份工作，需要靠这份工作来生存，娶妻生子，享受生活。

并不是所有的人都希望让Oracle成为自己生命的一部分，这是很现实的，不过我们虽然可以仅仅把Oracle当做是谋生手段，但也还是可以同时尝试了解Oracle更多的本质，像Oracle一样思考。

## &lt;&lt;DBA的思想天空&gt;&gt;

对大多数人而言，像Oracle一样思考虽然不能带给你更多的生活乐趣，但是通过以这样的方式去学习和思考，会更加精确地了解Oracle的精髓，让自己在DBA的成长过程中少走弯路。

10多年前我第一次接触Java的时候，感到十分头痛。

不是自夸，10多年前，我是一个相当不错的C程序员，最高纪录是一天之内编写500多行复杂的代码，而且一次性编译通过，一次性测试通过，这样的记录的诞生是基于十分良好的过程思维能力的。

不过我这个自封的编程高手第一次接触Java的时候，却感到十分吃力。

我无法用面向对象的思想去编写程序，所以学起Java来十分痛苦，几次学习，最后都放弃了。

直到有一天我看到了一本英文的书籍Thinking in Java，通过这本书，我掌握了Java和面向对象设计、编程的主要思路。

自从看了这本书之后，我再次面对Java程序的时候，发现一切都是那么地简单，很快我就掌握了Java编程。

现在我虽仍然还只是一个三流的Java程序员，不过粉丝网的一些修修补补的工作我完全能够胜任了，而且在和一些Java开发人员交流的时候，我也能够很快地理解他们的思路。

后来我总结了一下，在看Thinking in Java这本书之前，我在编写Java程序的时候，并没有理解面向对象编程的概念，只能是照猫画虎，拿着一个例子在上面修改。

实际上我的编程风格还是面向过程的，因此写出来的代码质量很差。

而通过阅读Thinking in Java，我终于学会了面向对象的方法，用Java本身的思想去考虑问题，因此能够更加准确地抓住问题的本质。

我想，学习Oracle数据库也是这样，如果我们通过一个又一个案例去学习Oracle，那么将永远停留在表层上。

有些DBA只能重复相同的案例，这样的DBA，哪怕干上10年20年，也可能只学到Oracle的一些皮毛，碰到一个没有见到的案例，可能就会感到手足无措。

而水平高一些的DBA往往能够判断出案例的相似性，并通过分析找到类似案例的解决方法，这其实就是因为透过现象看到了问题的本质。

很多DBA可能都碰到过我下面所说的一些问题，有时候我们无法评估某项调整可能对系统带来的影响，有时候我们面对一个复杂局面的时候很难快速找到问题的关键，也有时候我们在为解决某种等待事件而感到无从入手。

实际上，遇到这些问题，都是因为缺乏对于Oracle内部原理的充分认识。

在很多情况下，当经验无法为我们提供足够支持的时候，就必须从原理出发进行思考，才有可能真正掌握问题的根源，从而解决问题。

前几天我在一个客户现场做数据拯救工作的时候，他们的备库（也就是现在的主生产库）突然宕机了，当时客户正在做一个删除临时文件的操作，领导就认为他这个操作导致了宕机，而操作人员也觉得很冤枉，因为这是一个十分常规的操作。

我看了看日志文件，从日志上看不出任何由于临时表空间和临时段操作引起的问题，同时又看到了一个好像是人工操作停库的信息，于是推断可能是人为操作所致。

后来经过多方面查证，确实是有个DBA在家里远程做维护的时候，发现操作HANG住了，情急之下，直接重启了数据库。

如果你不了解临时段和临时表空间操作的原理，面对这个问题，很可能一上来就把重点放在删除临时文件导致宕机问题的分析上面，这样就偏离了正确的方向，解决问题的效率和成功率就会大大降低。

我们强调理论的重要性，也不是片面强调理论而不重视实践。

Oracle数据库是实践性很强的，没有实践，光学习理论是无法成为真正的高手。

比如说我们学习了很多OWI相关的理论，了解了数据库等待事件和一些状态指标的含义，但是我们看到一个库的AWR报告的时候，还是无法知道某个指标是否正常。

当对大型OLTP系统缺乏实践经验的时候，我们就无法知道大型OLTP系统的一些技术指标的特性，因此也很难从中找到疑点，进而找到解决问题的方法。

这些年里我接触过大量的DBA，我一般把他们分为四大类。

第一类DBA是经验型的，处理问题的主要方式取决于以往的经验，他们往往都有很好的习惯，会把每

## &lt;&lt;DBA的思想天空&gt;&gt;

一个处理过的案例整理出来，今后再碰到这类案例的时候，他们会很快地解决问题。

随着工作时间的增长，他们的技术也会相应地提高。

第二类DBA是理论型的，他们具有很深的理论基础，经常探讨一些“Oracle Internal Only”的高深问题，比如他们能够很清晰地告诉你共享池分配的算法，告诉你checkpoint的工作原理，但是这些DBA往往缺乏实际的工作经验，他们研究Oracle却很少有机会接触大型的数据库系统，因此实际解决问题的能力并不强。

另外，由于他们的知识比较片面，在某些方面很深入，而某些方面就是浅尝辄止，这种不均衡导致他们的知识只是以点的形式存在，无法串成整体，因此那些很深入的研究并不能给他们实际工作带来多大的帮助。

第三类DBA是技巧型的，他们并不注重理论的学习和经验的积累，在处理问题的时候往往能够利用Metalink和谷歌、百度之类的工具去搜索解决方案，这类DBA最为常见。

他们处理问题往往靠运气，而且一些他们自鸣得意的成功案例往往也是经不起推敲的，下回碰到类似案例时，可能还会失败。

第四类DBA是虚心请教型的，他们无论碰到什么问题，甚至连错误信息都没有看明白，就开始叫“我的系统出问题了”，然后到处去问如何解决。

实际上，这四类DBA都是有缺陷的。

第一类DBA可能经过多年的工作，有十分丰富的经验，处理问题的能力很强，而且分析问题十分敏感，很容易抓到问题的关键，但是由于没有深入理解Oracle的理论，碰到一些较为深入的问题的时候，就不容易立刻找到关键。

虽然凭借着自身丰富的经验和问题分析排查能力，他们最终也能解决大部分的问题，但是往往问题解决后还是没有真正弄明白为什么会解决问题，下一次碰到类似的问题，可能还是要花很大的代价。

第二类DBA在某些方面的理论知识很强，总是喜欢研究一些十分高深的原理性的东西，但是这类DBA的主要精力都放在了研究一些Oracle内部原理上了，他们没有很多的时间去实践他们学到的理论。

这类DBA往往知识面较为狭窄，仅精通于自己研究比较深入的领域，在实际工作中也很难发挥出自身的理论研究特长。

第三类DBA实际上在我们的现实生活中是最常见的，“万事不明问百度，百度不明就抓瞎”，确实谷歌、百度和Metalink能够帮助我们解决不少问题，但是这类DBA往往在问题解决后没有好好思考一下，为什么这个方法能够解决问题，更没有认真总结和归纳一下，下一次碰到类似的问题，还是无法依靠自己的思考去解决问题。

于是再Google一把，也许这一次运气没有那么好了，Google出来的资料不是上回的那个了，于是结果可能是很悲惨的。

第四类DBA在我们现实生活中也经常出现，网络社会通信十分发达，打个电话或者在qq群里、msn里问问，也许就有人帮忙解决问题。

久而久之，这些人放弃了自己的思考，碰到一点点小问题都要找人问。

缺乏独立思考问题能力的DBA，只能称为一个数据库操作员，实际上离真正的DBA还有十万八千里呢。

看到这里，大家可能明白了，老白实际上说的不是四类DBA，而是DBA的四种性格，这四种性格可能会集中在某一个人身上。

以老白学习DBA的经验来看，理论结合实践是十分重要的。

在2000年前，老白虽然做了很多项目，也是很多人眼里的Oracle数据库高手，但那时的老白就是第一类DBA的典型，没有经过多少理论学习，几乎所有的Oracle数据库的技能都是从实践中获得的。

虽然在实践中我总结出大量的经验，甚至有很多客户建议我写一本书，把我对Oracle的理解写出来，不过当我自信满满地开始写书的时候，却突然发现，我的一些知识需要进行确认，否则写出来就贻笑大方了。

于是我开始大量地学习Oracle的一些理论知识，随着写书过程的深入，我越发感到自身理论水平的不足。

《Oracle数据库深度历险》这本书我写了3年，实际上2002年就彻底放弃了出版这本书的念头，因为我

## &lt;&lt;DBA的思想天空&gt;&gt;

发现自己的理论知识确实还需要进一步的梳理。

但是我并没有放弃写作，因为我发现通过写作，我更为系统地将Oracle的理论知识梳理了一遍，这次梳理是通过我以前的知识体系、工作经验，与Oracle Concepts的理论基础进行了一次完整的整合。通过这3年的写作，我终于完全疏通了自己的Oracle的理论体系，好像一个练武术的人，终于打通了任督二脉，感到无比的畅快。

听老白说了这么一大通，是不是很多人都感觉到手脚发凉，难道成为一个合格的DBA有这么难吗？

如果我没有打通任督二脉，就不算一个合格的DBA吗？

实际上DBA成长的道路是很多的，并不一定要走老白这一条路，老白仅仅是根据自身的经历，通过这本书来帮助大家梳理Oracle的一些基础知识而已。

还是那句话，如果Oracle是你的爱好，那么你无论花多大代价去研究它都是值得的；如果Oracle只是你职场生涯中的一份工作而已，那么只要你认真对待它就可以了，没必要像老白那样执著。

作为一个DBA，理论学习和实践如何相结合是十分关键的。

在初期，一般来说DBA都是通过了某种途径接触了Oracle数据库，进行了一系列的操作。

在工作过程中发现了一些问题，才开始想到需要去看一些Oracle的书籍。

在这个阶段，Oracle官方文档的2days、7 days系列入门书籍就十分有效。

通过这些书籍你可以了解Oracle的一些基本的原理和基本的操作，帮你在工作中解决部分问题。

这样你在工作中就能够应对一些简单的问题了。

不过碰到稍微复杂一些的情况，你可能还是会发懵，这时，Oracle Concepts这本书就十分关键了。

从这个阶段开始看这本书是十分必要的，它有助于你在积累经验的过程中不断地完善理论。

不过，你可能还无法完全理解Oracle Concepts中的基本概念，通读这本书是十分必要的，但是不必要把每个问题都搞得十分清楚。

因为要达到这一点，你需要花费太多的时间和精力，同时也可能会由于缺乏足够的技术指导而无法真正理解问题的本质。

不过在这个阶段，碰到某些问题或者研究各种案例的时候，经常翻翻Oracle Concepts这本书是十分有益的，因为在处理问题的时候，你针对这个问题的思考会比较深入，这个时候，认真分析一下相关的理论是十分有效的。

对于处理过的每一件事情，都做一个比较详细的记录，是十分好的习惯。

记录下某个案例，可以供水平提高后再进行回顾，或者将案例提交给某个专家去评审，或者在网上和大家一起讨论，对于学习Oracle的原理都是十分好的方法，可以帮助你分析案例时提高对Oracle数据库原理的认知。

在这本书里，老白会把《Oracle数据库深度历险》中的一些内容，结合老白的实际工作经验展现给大家。

我会剖析原理，并结合案例来说明这些理论知识如何在实践中应用。

希望老白的这次写作经历，能够给大家带来一些帮助。

## <<DBA的思想天空>>

### 内容概要

数据库的性能优化一直是DBA日常工作中非常重要的组成部分，然而很多DBA在学习了大量技术，参加了大量培训后，仍然会在实际工作中遇到难以下手的问题。实际上，在数据库优化工作中，方法和思路远比技术实现重要得多。

本书重在介绍Oracle数据库的性能调优方法及相应的工作思路，但并不拘泥于技术细节。作者通过大量真实案例，深度剖析了相关技术原理，同时还阐述了理论知识在实践中的应用方法。优化工作的本质其实就是透过表象探寻根源，解决问题实现调优，正所谓“思路是道，操作方法是技”，得道是极大的提升，也是DBA的思想精髓。

## <<DBA的思想天空>>

### 作者简介

白鱈  
Oracle

ACE。

从事IT工作20年，曾供职于DEC、赛格计算机、长天集团、联想金融事业部等，担任过技术总监、应用体系总监等技术职务。

长期从事应用软件开发、设计与性能优化工作，1996年主持设计了国内首套电信级长话联机实时计费系统，荣获福建省科技进步三等奖；1998年主持设计了首套三检合一的检验检疫综合管理系统，荣获深圳市科技进步三等奖。

2002年起从事专业IT运维与技术支撑服务工作，在系统优化领域有十分丰富的工作经验，参与过数十个大型优化项目。

著有《Oracle 优化日记》、《Oracle RAC日记》等技术书籍。

储学荣

1992年毕业于南京大学计算机系，曾供职于得实集团、长天集团、联想集团等知名IT企业，担任程序员、软件架构师等职务。

从事过电信、金融、政府等行业核心系统研发工作，参与过UNIX内核开发工作并独立开发了类自然语言的电信计费专用语言ABC的编译器和P代码运行虚拟机。

2005年开始专门从事IT咨询与性能优化工作，在Oracle数据库性能优化方面有很深的造诣，并对Oracle数据库的内部结构有较深的研究，编写有大量的数据拯救工具。



## &lt;&lt;DBA的思想天空&gt;&gt;

## 书籍目录

## 目 录

## 第一部分 基础原理篇

## 第1章 理解Oracle数据库和实例

- 1.1 什么是Oracle数据库
- 1.2 Oracle数据库的物理结构
  - 1.2.1 Inventory
  - 1.2.2 口令文件
  - 1.2.3 参数文件
  - 1.2.4 控制文件
  - 1.2.5 在线日志文件
  - 1.2.6 数据文件
  - 1.2.7 归档日志文件
- 1.3 实例和多实例数据库
  - 1.3.1 什么是数据库实例
  - 1.3.2 多实例数据库
- 1.4 数据库后台进程
  - 1.4.1 进程结构
  - 1.4.2 后台进程的功能作介绍
  - 1.4.3 哪些后台进程可以杀
  - 1.4.4 是谁在执行SQL

## 第2章 理解DB Cache

- 2.1 什么是DB Cache
- 2.2 DB Cache的分配和DBWR的相关算法
  - 2.2.1 DB\_WRITER\_PROCESSES参数
  - 2.2.2 DB Cache的几个主要的链和CKPT算法
  - 2.2.3 检索某个DB BLOCK的模拟算法
- 2.3 DB Cache相关的参数锁和等待事件
- 2.4 DB Cache优化的一些探讨
  - 2.4.1 DB Cache和热块冲突
  - 2.4.2 使用KEEP POOL能改善CBC争用吗
  - 2.4.3 如何判断DB Cache是否足够
  - 2.4.4 DB Cache优化要点

## 第3章 理解共享池

- 3.1 共享池堆的内部结构
  - 3.1.1 进一步了解共享池
  - 3.1.2 共享池的子池技术
  - 3.1.3 字典缓存
  - 3.1.4 库缓存和游标
- 3.2 共享池和游标
  - 3.2.1 游标与游标共享
  - 3.2.2 游标与SQL的执行
  - 3.2.3 游标共享和绑定变量
  - 3.2.4 OPEN CURSOR和OPEN\_CURSORS参数
  - 3.2.5 CURSOR\_SPACE\_FOR\_TIME参数
  - 3.2.6 SESSION\_CACHED\_CURSORS参数和OPEN\_CURSORS

## &lt;&lt;DBA的思想天空&gt;&gt;

- 3.2.7 CURSOR\_SHARING和游标共享
- 3.2.8 游标的关闭
- 3.2.9 互斥锁和游标
- 3.3 共享池的相关参数
- 3.4 共享池故障处理
  - 3.4.1 著名的ORA-4031
  - 3.4.2 其他共享池常见故障
- 3.5 共享池优化的主要思路
- 第4章 理解控制文件
  - 4.1 控制文件的内部结构
    - 4.1.1 控制文件和控制文件事务
    - 4.1.2 控制文件自动扩展
    - 4.1.3 如何转储和分析控制文件
    - 4.1.4 文件头和控制文件信息
  - 4.2 故障处理和优化
    - 4.2.1 丢失或者损坏控制文件的处理方法
    - 4.2.2 控制文件的优化
- 第5章 理解REDO日志
  - 5.1 什么是REDO日志
  - 5.2 REDO的基本原理
    - 5.2.1 介质恢复和实例恢复的基本概念
    - 5.2.2 变化矢量和REDO记录
    - 5.2.3 日志缓冲和LGWR
    - 5.2.4 日志切换和REDO日志文件
    - 5.2.5 事务提交和回滚的过程
  - 5.3 REDO优化
    - 5.3.1 BULK操作能减少REDO吗
    - 5.3.2 如何优化LOG FILE SYNC等待事件
    - 5.3.3 SHUTDOWN ABORT无害吗
    - 5.3.4 关于REDO日志优化的建议
- 第6章 理解UNDO
  - 6.1 UNDO的基本原理
    - 6.1.1 UNDO表空间和回滚段
    - 6.1.2 ITL和UNDO
    - 6.1.3 如何转储UNDO
    - 6.1.4 UNDO自动管理是如何工作的
    - 6.1.5 系统回滚段的作用
    - 6.1.6 著名的ORA-1555
    - 6.1.7 回滚段手工管理
  - 6.2 如何分析和优化UNDO
- 第7章 理解PGA、临时表空间和排序
  - 7.1 基本概念
    - 7.1.1 临时表空间和临时段
    - 7.1.2 PGA和排序
    - 7.1.3 PGA和PGA\_AGGREGATE\_TARGET
    - 7.1.4 你应该知道的PGA自动管理内幕
  - 7.2 PGA优化的要点

## &lt;&lt;DBA的思想天空&gt;&gt;

## 第8章 理解ASM的结构

## 8.1 什么是ASM

## 8.2 ASM的结构

## 8.2.1 ASM DISKHEADER的结构

## 8.2.2 ASM FILE DIRECTORY文件结构

## 8.2.3 ASM ALIAS DIRECTORY文件结构

## 8.2.4 ASM DISK DIRECTORY文件结构

## 8.2.5 从ASM存储结构谈ASM日常维护的要点

## 8.3 如何使用KFED分析和修改ASM数据

## 8.4 如何使用AMDU导出ASM文件

## 第9章 理解数据块结构

## 9.1 理解数据块头结构

## 9.2 理解ITL

## 9.3 理解记录结构

## 9.4 解析Oracle字段的内部数据存储格式

## 9.5 理解LOB的存储结构

## 第10章 理解表的结构

## 10.1 到底什么是“表”

## 10.1.1 PCTFREE和行链

## 10.1.2 那些逝去的老参数

## 10.1.3 减少热块冲突的方法

## 10.2 从数据块结构看目前主流容灾技术

## 10.3 案例——简单任务

## 第11章 理解索引

## 11.1 反转键索引的误区

## 11.2 索引访问的方式

## 11.2.1 小表用索引有意义吗

## 11.2.2 位图索引为什么不适合大并发量环境

## 11.3 重建索引的作用

## 11.4 索引使用的“三大纪律八项注意”

## 11.5 案例——索引危机

## 第12章 理解分区表

## 12.1 什么是分区表

## 12.2 分区表对海量数据的意义

## 12.2.1 分区表和历史数据归档

## 12.2.2 分区表和高水位推进

## 12.2.3 分区表和RAC环境

## 12.2.4 分区主键和分区粒度的选择

## 第13章 理解序列

## 13.1 什么是序列

## 13.2 序列的使用和优化

## 第二部分 分析思路篇

## 第14章 问题分析综述

## 14.1 如何抓住蝴蝶效应中的那只蝴蝶

## 14.2 为什么要强调基础概念

## 14.3 工作中的好习惯带来的福利

## 第15章 DBA分析思路的探讨

## <<DBA的思想天空>>

- 15.1 问题分析总路线图
- 15.2 普通故障的分析路线
- 15.3 性能问题的分析路线
- 15.4 SQL语句的分析路线
- 15.5 利用你知道的原理缩小问题的范围
- 15.6 关闭问题的条件
- 15.7 灵活运用你的知识
- 15.8 DBA需要与时俱进
- 15.9 多表连接的优化技巧
- 15.10 理论如何联系实践
- 第三部分 典型案例篇
- 第16章 RAC故障分析
  - 16.1 LOG\_ARCHIVE\_MAX\_PROCESS导致的RAC脑裂
  - 16.2 RAC系统故障的处理过程
  - 16.3 三天两次严重故障
- 第17章 ORA-600故障
  - 17.1 ORA-600 [12700]错误的分析过程
  - 17.2 ORA-600 [kdsgrp1]的处理案例
- 第18章 性能问题分析
  - 18.1 压力测试遇到的问题
  - 18.2 IMP导入性能问题的分析
  - 18.3 并行操作为什么无法执行
- 第19章 SQL优化
  - 19.1 一个常用的SQL优化方法
  - 19.2 一个查找IP所属区域的SQL优化思路
- 结束语

## &lt;&lt;DBA的思想天空&gt;&gt;

## 章节摘录

版权页：插图：对于单实例的系统，实例恢复一般是在数据库实例异常故障后数据库重启时进行，当数据库执行了SHUTDOWN ABORT命令或者由于操作系统、主机等原因宕机重启后，在ALTER DATABASE OPEN时，就会自动进行实例恢复。

而在。

RAC环境中，如果某个实例宕掉了，活着的实例将会接管，替宕掉的实例做实例恢复。

除非是所有的实例都宕掉了，这样的话，第一个执行ALTER DATABASE OPEN的实例将会做实例恢复。

这也是REDO日志文件作为实例私有的组件必须存放在共享存储上的原因。

Oracle数据库的高速缓存机制是以性能为导向的。

高速缓存机制应该最大限度地提高数据库的性能，因此缓存被写入数据文件时总是尽可能推迟。

这种机制大大提高了数据库的性能，但是当实例出现故障时，可能存在一些问题。

首先，可能某些事物对数据文件的修改并没有完全写入磁盘，或者磁盘文件中丢失了某些已提交事务对数据文件的修改信息。

其次，可能某些还没有提交的事务对数据文件的修改已经被写入磁盘文件了。

也有可能某个原子变更有一部分数据已经被写入文件，而另外一部分数据还没有被写入磁盘文件。

实例恢复就是要通过ONLINE REDO LOG文件中记录的信息，自动完成上述数据的修复工作。

这个过程是完全自动的，不需要人工干预。

在这个机制里，有两个问题需要解决。

第一个是如何确保已经提交的事务不会丢失，第二个是如何在数据库性能和实例恢复所需要的时间上做出平衡，既确保数据库性能不会下降，又保证实例恢复可以快速进行。

## <<DBA的思想天空>>

### 编辑推荐

Oracle资深专家白鱻、储学荣联袂打造数据库力作以真实的案例贯穿《DBA的思想天空——感悟Oracle数据库本质》全书，剖析数据库技术原理带领读者感悟DBA思想精粹，揭示原理，但不是简单地炫耀内部原理，而是结合内部原理让DBA理解分析思路。

<<DBA的思想天空>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>