

<<品悟性能优化>>

图书基本信息

书名：<<品悟性能优化>>

13位ISBN编号：9787302251118

10位ISBN编号：7302251118

出版时间：2011-5

出版时间：清华大学出版社

作者：罗敏

页数：488

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>

## <<品悟性能优化>>

### 内容概要

在当前高速发展的信息时代，IT系统性能问题一直是横亘在广大IT人士面前的一座难以逾越的大山。

在数据库市场占有率最高的Oracle性能优化技术，更是业界各层次人士所关注的焦点技术之一。

本书基于作者多年在此领域耕耘的经验和体会，遵循Oracle公司总结的性能优化方法论，从需求分析、架构设计、数据库设计、应用设计和开发、运行维护等软件工程全生命周期的整体高度，去描述性能问题和相关优化技术，特别是强调了基础技术合理运用的重要性。

本书贯穿了作者多年在国内银行、电信、政府等行业所经历过的大量案例，通过案例佐证相关技术的运用是本书一大特色。

性能问题不仅仅是技术问题，性能管理甚至重于优化技术本身。

通过性能优化这样精细、缜密的工作，不仅能看到事物的本质和规律，更能让有心者对这个大千世界充满感慨。

以技术为平台，以轻松、调侃方式抒发作者对IT行业、社会的感悟，也是本书的鲜明风格。

## <<品悟性能优化>>

### 作者简介

罗敏，80年代毕业于武汉大学计算机科学系，国防科学技术大学计算机学院获得硕士学位。

自1988年开始Oracle技术的研究和应用开发工作，特别是在Oracle中国公司的10年时间里，分别在顾问咨询部、技术服务部担任资深技术顾问。

在银行、电信、政府等行业和部门参与了多个大型IT系统的建设，提供了体系结构设计、数据库设计、应用开发设计指导、性能优化、数据备份恢复、容灾系统建设、数据仓库系统建设、数据库安全性等领域的咨询和技术支持服务，并为国内主要软件开发商和集成商进行过多场Oracle高级技术应用培训和交流活动。

## &lt;&lt; 品悟性能优化 &gt;&gt;

## 书籍目录

## 第1章 从两个案例开始

- 1.1 关于案例的说明
- 1.2 体验日本人工作风格
  - 1.2.1 问题现象及解决过程
  - 1.2.2 日本人严谨、细致的工作风格
  - 1.2.3 日本人的IT投入观
  - 1.2.4 该系统的总体感觉
  - 1.2.5 在日本企业暖意洋洋的一幕
  - 1.2.6 也谈强国梦
- 1.3 国内某大型银行故障的解决
  - 1.3.1 天塌下来一样的故障
  - 1.3.2 故障原因其实很简单
  - 1.3.3 故障的启示
  - 1.3.4 2010年银行案例的进一步感悟

## 第2章 Oracle数据库性能优化方法论

- 2.1 关于性能优化的误区
  - 2.1.1 “你调了哪些参数”
  - 2.1.2 “性能优化主要是DBA和系统管理员的工作”
  - 2.1.3 “开发阶段无须太多考虑性能问题”
  - 2.1.4 “优化SQL，就是如何编写SQL”
  - 2.1.5 “多表连接性能太差”
  - 2.1.6 “CPU利用率越低越好”
  - 2.1.7 “大内存能解决性能问题”
  - 2.1.8 “性能分析就是分析低层细节”
- 2.2 性能优化过程——自顶向下
  - 2.2.1 为时已晚
  - 2.2.2 什么叫自顶向下方法论
  - 2.2.3 体验方法论
- 2.3 高质量IT系统的正确认识
  - 2.3.1 高质量IT系统的目标
  - 2.3.2 目标的综合平衡
  - 2.3.3 你只管进，不管出啊
- 2.4 20/80规则
  - 2.4.1 性能优化中也有20/80规则
  - 2.4.2 用数据诠释20/80规则
- 2.5 性能优化过程——自底向上
  - 2.5.1 什么叫自底向上方法论？
  - 2.5.2 客户要给我上课
- 2.6 性能优化中的角色分工
  - 2.6.1 老外的角色分工
  - 2.6.2 国内的角色分工
- 2.7 应用开发指导思想
  - 2.7.1 管理重于技术
  - 2.7.2 我听后，开心死了

## &lt;&lt;品悟性能优化&gt;&gt;

## 2.8 合理运用技术的重要性

2.8.1 联机事务处理系统 (OLTP) 和决策支持系统 (OLAP)

2.8.2 “你们Oracle给我们出一个开发规范和指南吧”

2.8.3 4分钟如何优化到1秒钟

## 第3章 性能优化分析基本工具的使用

## 3.1 性能优化中的量化分析

3.1.1 隔靴抓痒

3.1.2 SQL语句到底是怎么执行的

3.1.3 性能分析都分析哪些量化指标

## 3.2 工欲善其事，必先利其器

3.2.1 SQL量化分析和优化工具

3.2.2 Oracle有大量实用的小工具和命令

## 3.3 SQL语句到底是怎么被执行的

3.3.1 最经典的执行计划分析工具

3.3.2 这种老掉牙的东西，还用啊

3.3.3 10g新功能：DBMS\_XPLAN

## 3.4 如何配套使用SQL\*Trace和TKPROF

3.4.1 又一对老古董

3.4.2 其实功能非常强

3.4.3 报告分析比如何产生报告更重要

## 3.5 最常用的工具：Autotrace

## 3.6 一个洋“忽悠”的故事

3.6.1 洋和尚到中国来念梵文了

3.6.2 洋大“忽悠”啊

## 3.7 性能优化与“三个代表”

3.7.1 重温“三个代表”

3.7.2 案例背景

3.7.3 自底向上方法论的运用

3.7.4 关键应用问题的解决

3.7.5 诠释“三个代表”

## 第4章 基本索引的使用

## 4.1 索引既简单又复杂

4.1.1 关于索引的需求

4.1.2 索引其实好简单

4.1.3 索引其实好难

4.1.4 想做个懂Oracle索引的专家，难上加难

## 4.2 索引设计基本建议

4.2.1 Oracle索引长什么样

4.2.2 B\*树单字段索引设计建议

4.2.3 一招鲜，吃遍天

## 4.3 如何避免索引被抑制

4.3.1 无从下手，郁闷至极！

4.3.2 幸亏父母都是数学老师

4.3.3 慎用自定义函数

4.3.4 关于函数索引使用的建议

4.3.5 其实是数据库设计问题

## &lt;&lt; 品悟性能优化 &gt;&gt;

- 4.4 一把双刃剑：复合索引
  - 4.4.1 复合索引的重要性
  - 4.4.2 我如何“戏弄”客户
  - 4.4.3 复合索引原理和设计建议
  - 4.4.4 IT系统是面向客户的，不是给领导看的
- 4.5 一个既简单又复杂的故事
  - 4.5.1 女儿说我吹牛了
  - 4.5.2 故事上集
  - 4.5.3 故事中集
  - 4.5.4 故事下集
- 4.6 如何进行索引监控分析和优化
  - 4.6.1 为什么索引I/O那么高
  - 4.6.2 别乱建索引
  - 4.6.3 如何发现多余的索引
  - 4.6.4 如何进行索引碎片分析和整理
- 第5章 为应用软件设计更好的性能和可扩展性
  - 5.1 基本概念和原理
    - 5.1.1 本章标题有点大吧
    - 5.1.2 一个屡见不鲜的错误
    - 5.1.3 解剖SQL语句执行过程
  - 5.2 语句共享性原理
    - 5.2.1 再说联机事务处理系统（OLTP）和决策支持系统（OLAP）
    - 5.2.2 如何实现语句共享化
    - 5.2.3 开发人员永远比Oracle聪明
    - 5.2.4 技术服务工作，越做胆子越小
    - 5.2.5 如何量化评估语句共享性
  - 5.3 回到日本企业案例
    - 5.3.1 深入分析技术原因
    - 5.3.2 被日本人较真的滋味其实不好受
  - 5.4 语句共享性的深入分析
    - 5.4.1 语句共享性和查询统计系统的关系
    - 5.4.2 语句共享性与扩展性的关系
- 第6章 如何提高排序、表连接性能
  - 6.1 如何提高排序性能
    - 6.1.1 能不排序就不排序——废话一句
    - 6.1.2 查询欠费最高的前100名手机客户
    - 6.1.3 痛心疾首的一刻
    - 6.1.4 IBM和Oracle：亦敌亦友
  - 6.2 Oracle表连接技术和应用
    - 6.2.1 数据库精髓之一：表连接
    - 6.2.2 最经典、最常用的表连接技术——嵌套循环
    - 6.2.3 嵌套循环连接与索引
    - 6.2.4 嵌套循环连接的应用场景及效率
    - 6.2.5 适合于大批量数据处理的连接技术
  - 6.3 多表连接优化的基本思路
    - 6.3.1 总体思路
    - 6.3.2 OLTP应用的表连接优化

## &lt;&lt;品悟性能优化&gt;&gt;

## 6.4 如何使用子查询

## 6.4.1 使用子查询好不好

## 6.4.2 到底是使用in还是exists

## 6.5 回到20/80规则

## 6.5.1 优化详细过程

## 6.5.2 技术方面总结

## 6.5.3 每项工作做到最好都不容易

## 第7章 应用综合优化及总结

## 7.1 更多的优化案例

## 7.1.1 数据类型不一致导致的问题

## 7.1.2 多此一举的操作

## 7.1.3 错误使用HINT

## 7.1.4 Oracle和IBM又一次成功合作

## 7.2 可怕的笛卡儿乘积

## 7.2.1 问题的发生和初步解决

## 7.2.2 其实是设计和开发中更深层次问题

## 7.3 说说全表扫描

## 7.3.1 导致数据库性能问题的常见原因

## 7.3.2 何谓全表扫描

## 7.3.3 数据增长与全表扫描的关系

## 7.3.4 硬件太多了

## 7.3.5 导致技术运用复杂化的其他问题

## 7.3.6 更多的类比和感慨

## 7.4 导致性能问题的其他原因

## 7.5 一个应用软件的综合优化

## 7.5.1 优化前的状况

## 7.5.2 优化策略及分工合作

## 7.5.3 优化效果及原因分析

## 7.5.4 主管部门的反应

## 7.5.5 美妙的三降预言同时实现

## 7.5.6 优化工作的艰巨性和长期性

## 7.6 一个朴实无华的好系统

## 7.6.1 国人也能做出精良的好系统

## 7.6.2 巨大升值空间

## 7.6.3 瑕不掩瑜

## 第8章 Oracle分区技术及应用

## 8.1 硅谷之行

## 8.1.1 IT人的圣地：硅谷

## 8.1.2 我在Oracle总部中邪了

## 8.2 我对Oracle分区技术的认知过程

## 8.2.1 初尝分区甜头

## 8.2.2 分区给我的痛苦体验

## 8.2.3 全面理解分区技术

## 8.3 分区表技术

## 8.3.1 分区技术原理：分而治之

## 8.3.2 分区表技术概述

## 8.3.3 11g的分区新技术

## &lt;&lt;品悟性能优化&gt;&gt;

## 8.4 分区索引技术

8.4.1 分区索引技术好难哦

8.4.2 10分钟让你理解最难的分区索引

8.4.3 分区索引设计指南

## 8.5 更多的分区技术

8.5.1 一个神奇的分区技术

8.5.2 Oracle分区技术发展史

## 8.6 如何实施和评估分区

8.6.1 分区设计建议

8.6.2 分区效果评估

8.6.3 如何在生产系统实施分区

## 8.7 某行业分区方案设计的曲折过程

8.7.1 第一阶段：出师不利

8.7.2 第二阶段：经验主义错误

8.7.3 第三阶段：初见成效

8.7.4 第四阶段：日臻完善

## 8.8 分区方案中常见问题探讨

8.8.1 问题1：目标方面的误区

8.8.2 问题2：分区表设计方面的误区

8.8.3 问题3：没有充分考虑应用设计和开发的误区

8.8.4 问题4：分区表空间设计方面的误区

8.8.5 问题5：分区在大批量数据处理中的误区

8.8.6 问题6：分区索引设计方面的误区

8.8.7 无止境的分区技术

## 第9章 架构与性能优化

## 9.1 该谈谈架构了

9.1.1 架构与性能的关系

9.1.2 Oracle高端架构产品与性能的关系

9.1.3 Oracle架构的重要性

## 9.2 基本概念很重要

9.2.1 什么是Oracle数据库

9.2.2 服务器、实例和数据库的关系

9.2.3 关于架构方面的误区

## 9.3 IT系统架构现状分析

9.3.1 一副并不美妙的大蜘蛛网

9.3.2 现有体系结构特点分析

9.3.3 现有体系结构评估

## 9.4 Oracle网格计算

9.4.1 Oracle 10g = 网格计算

9.4.2 按网格计算设计数据库架构

9.4.3 乌托邦式架构就是好

9.4.4 初级阶段的设计建议

9.4.5 关于真正大集中的疑虑

## 9.5 云计算与性能优化

9.5.1 满天翻滚的云

9.5.2 Oracle的红云

9.5.3 客户关注的云计算话题



## &lt;&lt;品悟性能优化&gt;&gt;

## 9.6 Oracle数据库分布式架构

## 9.6.1 数据复制技术

## 9.6.2 Data Guard技术简介

## 9.6.3 Streams技术简介

## 9.6.4 其他数据同步技术

## 9.6.5 数据同步技术的定位和比较

## 9.7 我看分布式架构

## 9.7.1 我不喜欢分布式架构

## 9.7.2 数据大集中与分布式架构

## 9.7.3 分布式架构的用武之地

## 9.7.4 真正的返璞归真

## 9.8 誓做抗拒拆迁的刁民

## 9.8.1 拆迁大锤已高高举起

## 9.8.2 统一战线发挥重要作用

## 9.8.3 人民战争的汪洋大海

## 9.9 一个本来平淡的日子

## 9.9.1 什么专家，拿了钱就跑？

## 9.9.2 艰难的问题诊断过程

## 9.9.3 一根救命稻草

## 9.9.4 技术方面的教训和感悟

## 9.9.5 犹豫半天的话语

## 第10章 RAC与性能优化

## 10.1 关于RAC的一些误解和疑虑

## 10.2 RAC技术原理

## 10.2.1 系统介绍RAC架构和原理

## 10.2.2 RAC到底有什么好处

## 10.2.3 10g RAC架构新特性

## 10.2.4 我害怕Oracle什么技术工作

## 10.3 RAC架构的优势

## 10.3.1 为什么RAC架构比HA架构好

## 10.3.2 客户的方案不一定是最优的

## 10.4 RAC实施方法论

## 10.4.1 还是方法论重要

## 10.4.2 “你打个补丁要三天啊？”

## 10.4.3 如何降低RAC实施和运行风险

## 10.4.4 RAC其实背了好多黑锅

## 10.5 RAC性能优化原理

## 10.5.1 RAC性能优化等同于单事例

## 10.5.2 RAC性能问题与应用关系

## 10.5.3 应用在RAC环境下部署的最佳方式

## 10.6 RAC环境下的性能分析

## 10.6.1 RAC性能分析基本策略

## 10.6.2 AWR报告中的RAC性能分析

## 10.6.3 ADDM报告中的RAC问题原因分析

## 10.6.4 GCS性能分析

## &lt;&lt;品悟性能优化&gt;&gt;

- 10.6.5 GES性能分析
- 10.6.6 下得去，还要上得来
- 10.7 RAC高可用性
  - 10.7.1 RAC高可用性技术其实很复杂
  - 10.7.2 RAC高可用性实施思路
  - 10.7.3 RAC高可用性测试案例和测试过程
- 10.8 RAC可扩展性
  - 10.8.1 RAC不能超过4个节点？
  - 10.8.2 Oracle总部RAC专家的观点
  - 10.8.3 某大型交易系统的扩展性测试
  - 10.8.4 如何实施RAC扩展性
- 10.9 RAC运行维护和故障诊断
  - 10.9.1 RAC运行维护建议
  - 10.9.2 RAC故障诊断经验谈
  - 10.9.3 瞎猫碰上死耗子
- 第11章 数据仓库中的性能优化
  - 11.1 我看数据仓库
    - 11.1.1 数据仓库不是仓库管理软件
    - 11.1.2 数据仓库鼻祖的精确定义
    - 11.1.3 数据仓库的应用特点
    - 11.1.4 我所理解的数据仓库
    - 11.1.5 本书讲述的数据仓库
  - 11.2 数据仓库应用开发指导思想
    - 11.2.1 数据仓库应用开发指导思想建议
    - 11.2.2 案例为证
    - 11.2.3 如何贯彻大批量、并行处理？
  - 11.3 并行处理技术的应用
    - 11.3.1 Oracle并行处理技术无处不在
    - 11.3.2 并行处理举例
    - 11.3.3 并行技术的几个层面
    - 11.3.4 并行处理经验
    - 11.3.5 榨干所有硬件资源
  - 11.4 Oracle是个大计算器
    - 11.4.1 告别农耕时代
    - 11.4.2 Oracle不仅是一个存数据的大容器
  - 11.5 大批量数据ETL案例
    - 11.5.1 VIP客户判断标准
    - 11.5.2 外部表
    - 11.5.3 MERGE语句
    - 11.5.4 VIP计算总体流程图
    - 11.5.5 简述一个流程
    - 11.5.6 方案评估
  - 11.6 “非典”期间的一个典型性问题
    - 11.6.1 一个应用开发中的典型性问题
    - 11.6.2 Oracle系统级临时表

## &lt;&lt; 品悟性能优化 &gt;&gt;

- 11.7 一种快速高效的数据仓库加载方案
  - 11.7.1 让洋鬼子激动地蹦到桌子上去
  - 11.7.2 快速高效的数据仓库加载方案
- 11.8 报表优化技术
  - 11.8.1 我的第一次软件开发经历
  - 11.8.2 现在的报表处理状况
  - 11.8.3 报表优化核心技术：物化视图和语句重写
  - 11.8.4 为什么不要自己编写汇总表
  - 11.8.5 报表优化的基本思路及示例
  - 11.8.6 报表优化示例
  - 11.8.7 为什么没有实现语句重写
  - 11.8.8 IT行业到底是买方市场还是卖方市场
- 第12章 统计信息采集与性能优化
  - 12.1 我闯大祸了
    - 12.1.1 常在河边走，哪有不湿鞋的
    - 12.1.2 太急于表现了
    - 12.1.3 建一个索引，搞死一个系统
    - 12.1.4 我被骂得满地找地缝
    - 12.1.5 初识问题原因
    - 12.1.6 问题根本原因
  - 12.2 优化器原理和统计信息采集作用
    - 12.2.1 SQL语句执行过程
    - 12.2.2 基于规则优化器（RBO）简介
    - 12.2.3 基于成本优化器（CBO）简介
    - 12.2.4 如何将葫芦和瓢都按下
    - 12.2.5 为什么要进行统计信息采集
  - 12.3 自动采集统计信息
    - 12.3.1 自动采集统计信息的特点
    - 12.3.2 自动还是手工
    - 12.3.3 超长的自动统计信息采集
    - 12.3.4 一次变味的数据库升级技术研讨会
  - 12.4 定制采集统计信息
    - 12.4.1 统计信息采集基本策略
    - 12.4.2 统计信息采集实施策略
    - 12.4.3 统计信息采集具体方法
    - 12.4.4 Oracle 10g鬼精鬼精的
  - 12.5 若干最佳实践经验
    - 12.5.1 自动和手工结合进行统计信息采集
    - 12.5.2 锁住统计信息采集
    - 12.5.3 数据分布统计（Histogram）建议
    - 12.5.4 批处理中的统计信息采集
    - 12.5.5 铁路警察，各管一段
- 第13章 感悟性能优化分析的高级工具
  - 13.1 Oracle 10g = Oracle 10a
    - 13.1.1 外部手工管理变内部自动管理
    - 13.1.2 Oracle 10g都有哪些自动的东西
  - 13.2 AWR是个好东西

## &lt;&lt;品悟性能优化&gt;&gt;

- 13.2.1 AWR原理
- 13.2.2 AWR基本操作
- 13.2.3 把AWR功能用个够
- 13.3 ADDM : Oracle能自动诊断监控吗
  - 13.3.1 ADDM能干啥
  - 13.3.2 DBA要失业了吗
- 13.4 SQL优化进入工业化时代
  - 13.4.1 传统模式到工业化
  - 13.4.2 SQL Tuning Advisor能做哪些优化
  - 13.4.3 SQL Access Advisor能做哪些优化
  - 13.4.4 SQL Tuning Advisor和SQL Access Advisor的差异
  - 13.4.5 OEM中的每条命令我都会敲
  - 13.4.6 IT工业化时代的初级阶段
- 第14章 参数配置与性能优化
  - 14.1 神奇的“魔术师”
    - 14.1.1 不调系统参数
    - 14.1.2 调错系统参数
  - 14.2 漫谈初始化参数
    - 14.2.1 神奇的初始化参数
    - 14.2.2 参数设置基本思路和经验
    - 14.2.3 将Log Buffer设它个几百兆
  - 14.3 自动内存管理
    - 14.3.1 DBA真地快没活干了
    - 14.3.2 自动内存管理技术管用吗
  - 14.4 Buffer Cache优化
    - 14.4.1 Buffer Cache参数设置思路
    - 14.4.2 Buffer Cache优化技术
    - 14.4.3 Buffer Cache的其他优化技术
  - 14.5 Shared Pool优化
    - 14.5.1 Shared Pool参数设置思路
    - 14.5.2 Shared Pool优化技术
    - 14.5.3 再撞一次墙
  - 14.6 PGA优化
    - 14.6.1 PGA参数设置思路
    - 14.6.2 匪夷所思的做法
  - 14.7 奇妙的Oracle内部参数
    - 14.7.1 Oracle有神奇的内部参数吗？
    - 14.7.2 内部参数的一次神奇作用
    - 14.7.3 不要滥用内部参数
    - 14.7.4 令人眼花缭乱的内部参数和event
- 第15章 存储技术与性能优化
  - 15.1 什么时候才考虑I/O优化
  - 15.2 裸设备？
- 文件系统？
- ASM？

## &lt;&lt; 品悟性能优化 &gt;&gt;

15.2.1 裸设备有那么神奇吗？

15.2.2 客户期望值太高了

15.2.3 文件系统和裸设备的原理分析

15.2.4 文件系统同样好

15.3 RAID与性能优化

15.3.1 也说RAID

15.3.2 IBM太慷慨了

15.4 Oracle管存储了

15.4.1 关于ASM的疑虑

15.4.2 ASM是什么？

15.4.3 ASM有什么技术优势？

15.4.4 Oracle即将不支持裸设备了

15.5 ASM优化策略

15.5.1 裸设备与ASM的性能对比

15.5.2 如何保证ASM实施的高性能

15.5.3 ASM事例的参数设置建议

15.6 ASM实施案例

15.6.1 我所见过的真正海量数据库

15.6.2 裸设备还是ASM文件系统？

15.6.3 ASM、OMF、大表空间技术的完美结合

15.6.4 年轻教授被ASM气得直跳脚

15.6.5 宽容平和的心态

第16章 数据保护中的性能优化

16.1 Oracle丰富的数据保护技术

16.1.1 Oracle高可用性解决方案全景图

16.1.2 相关技术和产品的定位

16.2 RMAN实施现状分析

16.2.1 备份恢复只是磁带库厂商的事情？

16.2.2 RMAN实施中常见问题分析

16.3 RMAN备份的优化

16.3.1 RMAN备份优化的基本策略

16.3.2 RMAN备份优化的传统技术

16.3.3 在表空间级进行RMAN备份

16.3.4 10g的快速增量备份技术

16.3.5 备份压缩技术

16.4 RMAN恢复的优化

16.4.1 头疼医头，脚疼医脚

16.4.2 降低日志恢复量

16.4.3 增量更新备份

16.5 快速恢复数据的新技术：Flashback

16.5.1 人为逻辑错误是最大的单一因素

16.5.2 传统的数据恢复技术及缺陷

## &lt;&lt;品悟性能优化&gt;&gt;

- 16.5.3 Flashback技术概述
- 16.5.4 Flashback技术综合对比
- 16.5.5 Flashback技术与传统数据恢复技术综合运用
- 16.6 Data Guard实施中的优化
  - 16.6.1 容灾系统与生产系统是紧密相关的
  - 16.6.2 还是原理最重要
  - 16.6.3 日志传输的优化
  - 16.6.4 日志恢复的优化
  - 16.6.5 容灾系统与生产系统的配置关系
  - 16.6.6 也谈Data Guard与硬件存储镜像技术
  - 16.6.7 Data Guard和存储镜像技术的综合
- 第17章 故障诊断与性能优化
  - 17.1 故障诊断与性能优化的区别
    - 17.1.1 故障诊断与性能优化不完全是一回事
    - 17.1.2 故障诊断需要一个伟大的心脏
  - 17.2 大汗淋漓的故障诊断
    - 17.2.1 一个“Ctrl + C”几乎搞死一个系统
    - 17.2.2 啼笑皆非的故障处理过程
    - 17.2.3 胁从犯的自责
  - 17.3 可别小看数据坏块处理
    - 17.3.1 “关于Oracle腐败问题的处理”
    - 17.3.2 飞机落地了，资料还未看完
    - 17.3.3 收集信息、制定处理方案最重要
    - 17.3.4 数据坏块处理的八卦图
    - 17.3.5 别乱用DUL
    - 17.3.6 如何防范数据坏块
  - 17.4 堪比好莱坞大片的情节
    - 17.4.1 我的女同事被吓坏了
    - 17.4.2 惊心动魄的时刻！
    - 17.4.3 事件远没有结束
    - 17.4.4 其实原因很简单
  - 17.5 Oracle Buuuuuuuuuuuuuug
    - 17.5.1 我看Oracle Bug
    - 17.5.2 手工作坊与大工厂的差别
    - 17.5.3 一个展板都画不下的流程图
  - 17.6 软件版本管理和补丁实施
    - 17.6.1 相关术语和概念
    - 17.6.2 未雨绸缪的补丁实施计划
    - 17.6.3 打补丁那点事
    - 17.6.4 补丁冲突分析像侦探推理
- 第18章 DBA职责及性能管理
  - 18.1 我的专职DBA经历
    - 18.1.1 不太安心的“DBA”
    - 18.1.2 无所事事的“DBA”
    - 18.1.3 手忙脚乱的“DBA”
    - 18.1.4 无所事事的DBA

## &lt;&lt;品悟性能优化&gt;&gt;

- 18.2 DBA职责建议
  - 18.2.1 DBA的十大任务
  - 18.2.2 DBA的工作比例
  - 18.2.3 不太懂SQL的DBA
  - 18.2.4 一位技术实力超强的DBA
- 18.3 DBA在性能方面的工作
  - 18.3.1 每日的工作
  - 18.3.2 每周的工作
  - 18.3.3 每月的工作
  - 18.3.4 其他的工作
- 18.4 性能管理更重要
  - 18.4.1 性能问题其实是管理问题
  - 18.4.2 开发人员永远都长不大？
  
  - 18.4.3 Oracle核心技术开发团队的故事
- 18.5 开发与运行维护的脱节
  - 18.5.1 开发与运行维护部门的独立性
  - 18.5.2 开发与运行维护工作的脱节
- 18.6 客户/开发商/Oracle的分工合作
  - 18.6.1 目前的分工和定位
  - 18.6.2 客户在IT系统中的作用
  - 18.6.3 建议的分工和定位
  - 18.6.4 包含3种角色的项目组
- 18.7 分工合作的成功案例
  - 18.7.1 系统运行情况
  - 18.7.2 系统主要技术特点
  - 18.7.3 项目成功因素分析
  - 18.7.4 Oracle公司的服务经验
  - 18.7.5 项目的不足
- 第19章 软件就是服务
  - 19.1 Oracle服务体系概述
    - 19.1.1 Oracle公司组织结构一瞥
    - 19.1.2 Oracle丰富的服务产品
  - 19.2 我看Oracle标准服务
    - 19.2.1 标准服务不仅仅是法律条款
    - 19.2.2 标准服务的益处
  - 19.3 爱不释手的Metalink
    - 19.3.1 幸亏有Metalink
    - 19.3.2 初尝Oracle服务甜头
    - 19.3.3 Metalink是个大宝藏
    - 19.3.4 Metalink是个自助式的知识库
    - 19.3.5 在Metalink中提交SR的经验
    - 19.3.6 把Metalink当成学习工具
  - 19.4 Oracle高级客户服务
    - 19.4.1 ACS服务概述
    - 19.4.2 基于ITIL理念的ACS服务
    - 19.4.3 我们不是钟点工

## &lt;&lt;品悟性能优化&gt;&gt;

- 19.4.4 ACS的运行维护服务
- 19.4.5 IT系统挑战和ACS解决方案服务
- 19.4.6 几种ACS解决方案服务
- 19.5 又一次救火之后的感慨
  - 19.5.1 又着火了
  - 19.5.2 再次感谢Metalink
  - 19.5.3 客户把系统重新安装了
  - 19.5.4 其实还是服务问题
- 第20章 一个更全面的案例
  - 20.1 为升级而来
    - 20.1.1 初识客户
    - 20.1.2 升级方案遇到阻力
  - 20.2 以性能优化开路
    - 20.2.1 性能是升级的第一大风险
    - 20.2.2 调整服务思路
  - 20.3 性能整体评估
    - 20.3.1 先看操作系统数据
    - 20.3.2 数据库基准指标的采集
    - 20.3.3 性能分析策略和原则
  - 20.4 若干典型问题
    - 20.4.1 还是索引这样基础的问题
    - 20.4.2 发现了最大的性能瓶颈
    - 20.4.3 我把开发人员吓住了
    - 20.4.4 参数可调的余地太小
  - 20.5 难以解决的问题：中间表
    - 20.5.1 又一类典型问题
    - 20.5.2 9i没有合适的招
    - 20.5.3 10g的有效解决办法
  - 20.6 又说分区方案设计
    - 20.6.1 分区表太多了
    - 20.6.2 分区设计的其他问题
    - 20.6.3 综合平衡考虑问题不简单
  - 20.7 再说升级
    - 20.7.1 对升级的两种极端看法
    - 20.7.2 为什么要升级
    - 20.7.3 常见的升级方法
    - 20.7.4 Oracle升级服务包
    - 20.7.5 如何降低性能风险
- 第21章 综合类
  - 21.1 Oracle出硬件了
    - 21.1.1 我快变成硬件工程师了
    - 21.1.2 我看Exadata
    - 21.1.3 ACS在Exadata方面的服务
  - 21.2 Oracle全文检索技术
    - 21.2.1 Oracle能做搜索引擎
    - 21.2.2 茅塞顿开的解决方案
    - 21.2.3 林子大了，什么鸟都有



## <<品悟性能优化>>

- 21.3 什么是IT系统最宝贵的财富
  - 21.3.1 IT系统最宝贵的财富是信息本身
  - 21.3.2 也谈信息中心的作用
- 21.4 如何阅读Oracle联机文档
  - 21.4.1 Oracle联机文档的确是个宝藏
  - 21.4.2 合理分类阅读和利用
  - 21.4.3 按工作角色和任务去阅读
- 21.5 IT行业中的“伪”科学
  - 21.5.1 什么叫IT“伪”科学
  - 21.5.2 费力不讨好的事情
  - 21.5.3 我也是“伪”IT科学的吹鼓手
- 21.6 性能优化与桥牌
  - 21.6.1 大局观的重要
  - 21.6.2 实施计划的重要性
  - 21.6.3 应善于捕捉、利用信息
  - 21.6.4 合作、沟通的重要性
  - 21.6.5 简简单单，平平淡淡就是真
- 21.7 IT业其实还是个孩子
- 21.8 大话南游记

结束语

参考文献

推荐语

推荐语1

推荐语2

推荐语3

## &lt;&lt; 品悟性能优化 &gt;&gt;

## 章节摘录

版权页：插图：其实，Oracle性能优化方法论不是step\_by-step的设计开发和实施指南，上图就可以叫做性能优化方法论。

读者需要根据自己的知识、经验去体验之，并指导自己的设计、开发和实施工作。

例如：(1)优化工作开始得越早，其效益也越高。

同时，其付出的成本也最小。

在设计阶段就开始不仅考虑业务功能的实现，而且考虑实现的成本和代价，将是效益最高的。

(2)投产后才发现的问题有可能是灾难性的。

试想：如果在上线投产之后，才发现性能问题是因为没有遵循数据库规范化设计指导思想，因此将不得不重新修改表结构，也将重新进行相关应用程序的改造，重新进行各种功能测试、单元测试、集成测试等，付出的时间和代价可想而知。

(3)再好的硬件解决不了应用软件设计和开发的问题。

如果应用不合理而导致错误的执行路径，就如同从国贸去机场，本来走三环上机场路最快，如果非要绕道大兴过去，即便你坐的是奔驰，肯定也没我的伊兰特快。

(4)千万别将性能优化全部寄希望于硬件和系统层面。

Oracle上述原理图，应消除很多客户的过高期望：“赶紧把我的数据库系统由文件系统转为裸设备吧，据说性能能提高好几倍。

”至少本人这么多年从未听说过类似的成功案例。

说到存储技术，本人的观点是：文件系统、裸设备和ASM系统等各有优缺点，没有一种技术是十全十美的。

关键是在了解相关技术的基础上，充分考虑应用系统的不同需求，而选用最合适的存储技术。

后面有章节将专门讨论。

## &lt;&lt;品悟性能优化&gt;&gt;

## 媒体关注与评论

书中生动叙述的大量故事，以及对IT行业.社会的大量积极向上、也充满一定忧患意识的有感而发，我想也是本书吸引读者的一大亮点.....我很欣赏本书的主题和定位：领悟IT技术，品味人间百态。

——中国工程院院士：李德毅作为在Oracle中国公司服务部门就职多年的一名资深技术顾问，作者集多年驰骋在国内多个行业的经历和经验，“十年磨一剑”，以性能优化这一最容易引起业界各层次人士关注的话题为切入点，阐述了自己在架构、设计、开发、运维，甚至IT系统建设理念、文化等诸多方面的思考和独到见解，的确能引起IT从业人员的深思并受益。

——Oracle公司中国区客户服务部高级总监：田超（charles TIAN）《品悟性能优化》以数据库性能调优作为切入点，通过对案例故事事实景的生动形象论述，介绍了数据库建设和运维优化的方法论，叙述议论结合，管理技术兼备，语言风趣流畅。

作者专业技术积累深厚，善于总结提炼升华，简洁生动描述复杂问题背后难以忽视的真相”。

——国家税务总局电子税务中心副主任：陈梦林作为老朋友，翻阅此书，能感受到一个跃然纸上、鲜活而立体的作者：一位对先进IT技术充满激情、渴望和执着追求的技术专家；一位对客户富有责任心、称职得有些“越位”的IT服务人员；一位真性情，善于将技术观点寓于生动故事中的传道者。

——中国人民银行清算总中心副总工程师：贝劲松第一次看到枯燥的Oracle技术能以大量案例形式，以轻松、调侃、诙谐的语气写成这样，很有意思，也值得一看。

还有与作者的好多同感：在IT系统中，技术与管理并重；应用设计、开发其实对IT系统质量影响最大；基础技术的运用很重要..... ——山东移动通信有限公司首席DBA OCM认证大师：曹璐

## <<品悟性能优化>>

### 编辑推荐

《品悟性能优化》：生动的案例实景，严谨的技术描述，鲜活的人生感悟，轻松、调侃、诙谐的语言

。

<<品悟性能优化>>

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>