

深入解析Oracle体系架构和优化机制，以开放式可扩展的自动化运维体系为蓝本，原创自动化运维技术体系

数据库运维工作按需自动化，保障数据库高性能运行的同时，大大降低数据库运维成本，提高数据库管理工作效率

High-Performance and Automated  
Operation for Oracle

# Oracle高性能 自动化运维



冷波 著



机械工业出版社  
China Machine Press



数据库 技术丛书

High-Performance and Automated  
Operation for Oracle

# Oracle高性能 自动化运维



冷波 著



机械工业出版社  
China Machine Press



## 图书在版编目 (CIP) 数据

Oracle 高性能自动化运维 / 冷波著. —北京: 机械工业出版社, 2017.7  
(数据库技术丛书)

ISBN 978-7-111-57339-5

I. O… II. 冷… III. 关系数据库系统 IV. TP311.138

中国版本图书馆 CIP 数据核字 (2017) 第 142784 号

## Oracle 高性能自动化运维

---

出版发行: 机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码: 100037)

责任编辑: 何欣阳

责任校对: 殷虹

印 刷: 三河市宏图印务有限公司

版 次: 2017 年 7 月第 1 版第 1 次印刷

开 本: 186mm × 240mm 1/16

印 张: 24.5

书 号: ISBN 978-7-111-57339-5

定 价: 79.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88379426 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzit@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光 / 邹晓东



## 为什么要写这本书

2008 年的某一天，我怀着激动的心情参加了支付宝公司的面试，这次面试让我对国内最前沿的数据库自动化运维技术有了初步的认识，对我的职业生涯影响非常大，时刻激励着我以后不管遇到什么困难或者诱惑，Oracle 始终都是我追逐的爱好和目标。

在接下来几年的工作中，我较为深入地研究了 C 语言数据结构、Linux 操作系统原理以及 Shell（PHP/Java）编程等知识，这些技术能够很好地支撑 Oracle 运维管理。同时，多年的经验告诉我，对集运维与开发于一体的 Oracle 运维管理体系而言，精通一门开发语言是相当有必要的，因为只有这样才能更为深入地了解数据库与业务程序之间的架构设计，并能够更为精确地把控 Oracle 管理优化，从而在面对各种疑难问题时找到解决的突破口。

2014 年，我有幸与来自支付宝开发团队和淘宝运维团队的伙伴共事，学习交流了来自阿里的系统架构设计及运维管理理念，这使我编写自动化数据库运维书籍的想法更加强烈，并希望通过书籍与大家分享自己多年的工作经验。

## 本书主要特色

由于 Oracle 运维管理的专业性门槛较高，导致 Oracle 运维管理成本增加。在这种情况下，如何深入掌握 Oracle，如何提高数据库运维效率成为许多数据库管理者的困扰。尽管国内有关 Oracle 方面的书籍繁多，但却没有一本真正意义上介绍 Oracle 数据库运维自动化的书。本书从与 Oracle 紧密相关的操作系统层面入手，将操作系统架构与 Oracle 体系有机整合，以逐步深入的方式将 Oracle 基础理论运用到实际操作中，引导读者在面对故障时找到处理问题的正确方法。同时，结合开放式可扩展的自动化运维体系，将数据库运维工作按需自动化，在保障数据库高性能运行的同时，大大提高了数据库运维工作的效率。



## 本书读者对象

根据本书的内容特点，可以将阅读对象分为以下几类：

- ☐ Oracle 数据库管理人员；
- ☐ Oracle 数据库开发人员；
- ☐ 应用架构师；
- ☐ 数据库架构师；
- ☐ 系统集成架构师；
- ☐ 运维管理维护人员；
- ☐ 数据库技术爱好者。

## 如何阅读本书

阅读本书前需要对 Oracle 基础知识有一定的了解和掌握，只有这样才能保证阅读的畅通性。如果对某些知识点存在困惑，可以查阅 Oracle 在线官方文档（<http://docs.oracle.com>）获取帮助。

本书共 10 章，分为三篇：基础篇、优化篇、方法篇。

基础篇（第 1~4 章）主要介绍了 Linux 环境下 Oracle 内存体系架构的组成和功能，同时对 Oracle 日志与回滚段内容进行了讲解，在帮助读者深入理解 Oracle 知识体系的同时，为后续数据库优化及备份恢复打下基础。

优化篇（第 5~6 章）主要介绍了 Oracle SQL 优化的原理和思想，并结合 CBO 优化器对 Oracle 优化进行探索。其中，SQL 优化原理和思想是 CBO 优化的基础，将两者结合即可提供 Oracle 数据库优化参考。

方法篇（第 7~10 章）主要讲解了 Oracle 备份恢复的原理，以及如何制订符合自身数据库特点的 Oracle 备份恢复计划。同时，结合自动化运维管理系统、自动化历史分析系统及自动化实时监控系統对数据库自动化运维进行实践，帮助数据库维护人员从大量重复烦琐的运维工作中解放出来，实现高效率、自动化的数据库管理。

## 勘误和资源

由于本人水平有限，编写时间也比较仓促，书中难免有错误或者不准确的地方，在此恳请读者朋友批评指正。你可以将书中的错误发送到 Bug 勘误表页面中，同时，书中的源码文件也会发布到华章公司的网站上并及时更新。如果有任何疑问或者建议，也欢迎发送邮件到我的邮箱 [lbyouran@163.com](mailto:lbyouran@163.com)，期待你们真挚的意见反馈。



## 致谢

感谢 Oracle 技术社区的支持以及来自 BAT 等企业的同事和朋友们的技术分享，感谢 Oracle 小筑以及 Unix DBA 等技术社区的朋友们，他们包括但不限于李培跃、丁铁球、周操、文智辉、许剑锋、陶卫、贺学兵、周立明、李杰斌、罗炳森、龚明全、蹇波等。

感谢机械工业出版社华章公司的杨福川老师和李艺老师，你们的专业和热情深深地打动了我，让我能够坚持完成本书的编写。同时感谢你们对本书页面排版等方面的建议和支持。

感谢公司的同事和领导，谢谢你们给予我较为充裕的时间来完成本书。

特别感谢我的朋友李亚、李纯香、郑勇斌，谢谢你们对本书的大力支持。需要额外感谢的还有盖国强老师以及冯大辉老师等。

最后要感谢我的妻子刘杨及父母对我和孩子的照顾，因为有了你们的理解和支持，才使我有充足的时间和精力来完成本书。

谨以本书献给那些还在追逐、热爱 Oracle 技术，并依然坚持梦想的朋友们。

冷 菠



# Contents 目 录

## 前言

## 第一篇 基础篇

### 第 1 章 Linux 下的 Oracle..... 2

- 1.1 Linux 简介..... 2
- 1.2 Oracle 简介..... 3
- 1.3 Linux 内存体系的优势..... 3
- 1.4 Linux 内存体系与 Oracle 内存空间... 5
  - 1.4.1 Linux 用户空间与内核空间..... 5
  - 1.4.2 Linux 下的 Oracle 内存体系  
结构..... 5
  - 1.4.3 Linux 下的 Oracle 内存分配..... 6
- 1.5 小结..... 9

### 第 2 章 Oracle 内存体系结构..... 10

- 2.1 闩 (Latch)..... 10
  - 2.1.1 Latch 简介..... 10
  - 2.1.2 Latch Level..... 11
  - 2.1.3 Latch 获取模式..... 12
  - 2.1.4 Latch 获取等待..... 13
  - 2.1.5 Latch 资源清理回收..... 15

- 2.2 队列锁 (Enqueue Lock)..... 16
  - 2.2.1 Lock 与 Latch 的区别..... 16
  - 2.2.2 常见的 Lock..... 17
  - 2.2.3 Lock 相关参数..... 18
  - 2.2.4 Lock 先请求先服务机制..... 18
- 2.3 Library Cache..... 21
  - 2.3.1 Library Cache 与 SQL 游标..... 21
  - 2.3.2 Library Cache 内存结构..... 28
  - 2.3.3 Library Cache Lock (Pin)..... 31
- 2.4 Buffer Cache..... 34
  - 2.4.1 Buffer (Cache) Pool..... 34
  - 2.4.2 Cache Buffer Chain (Latch) ... 37
  - 2.4.3 Cache Buffer Pin..... 40
  - 2.4.4 Buffer Cache 等待与优化..... 44
- 2.5 小结..... 47

### 第 3 章 Oracle 重做日志 (Redo)..... 48

- 3.1 Redo 功能用途..... 48
- 3.2 Redo 组成结构..... 49
  - 3.2.1 Redo Header..... 49
  - 3.2.2 Redo Record..... 49
- 3.3 Redo 产生场景..... 53



|                                   |    |
|-----------------------------------|----|
| 3.3.1 Redo 与 DML 事务 .....         | 53 |
| 3.3.2 Redo 与 Block Cleanout ..... | 55 |
| 3.3.3 Redo 与 Block Write .....    | 56 |
| 3.3.4 Redo 与 Hot Backup .....     | 56 |
| 3.3.5 Redo 与 Direct Load .....    | 56 |
| 3.3.6 Redo 与 Nologging .....      | 57 |
| 3.4 Redo 优化 .....                 | 58 |
| 3.5 小结 .....                      | 61 |

## 第 4 章 Oracle 事务与回滚段 .....

|                                    |    |
|------------------------------------|----|
| 4.1 Oracle 事务 .....                | 62 |
| 4.1.1 Oracle 事务概览 .....            | 62 |
| 4.1.2 Oracle 事务 ACID 原则 .....      | 63 |
| 4.1.3 Oracle 事务与回滚段运行<br>机制 .....  | 64 |
| 4.2 Oracle 回滚段 .....               | 70 |
| 4.2.1 回滚段与一致性读 .....               | 70 |
| 4.2.2 回滚段与事务锁定 .....               | 71 |
| 4.2.3 回滚段与块清除 .....                | 72 |
| 4.3 Oracle 事务恢复 .....              | 73 |
| 4.3.1 回滚操作下的事务恢复 .....             | 73 |
| 4.3.2 进程崩溃下的事务恢复 .....             | 74 |
| 4.3.3 实例崩溃下的事务恢复 .....             | 75 |
| 4.3.4 数据库异常关闭下的事务<br>恢复 .....      | 75 |
| 4.4 Oracle 回滚段特殊恢复 .....           | 76 |
| 4.4.1 Oracle 回滚段特殊恢复隐藏<br>参数 ..... | 77 |
| 4.4.2 Oracle 回滚段特殊恢复场景 .....       | 78 |
| 4.4.3 Oracle 回滚段特殊恢复实战 .....       | 80 |
| 4.5 小结 .....                       | 81 |

## 第二篇 优化篇

## 第 5 章 Oracle SQL 优化 .....

|                                       |     |
|---------------------------------------|-----|
| 5.1 Oracle SQL 游标 .....               | 84  |
| 5.1.1 私有 SQL 游标 .....                 | 84  |
| 5.1.2 共享 SQL 游标 .....                 | 85  |
| 5.1.3 Library Cache 中的 SQL 游标 .....   | 85  |
| 5.1.4 SQL 游标与 Session 游标<br>缓存区 ..... | 88  |
| 5.2 Oracle SQL 解析与执行 .....            | 91  |
| 5.2.1 Oracle SQL 解析 .....             | 91  |
| 5.2.2 Oracle SQL 执行 .....             | 93  |
| 5.3 Oracle 表连接查询 .....                | 96  |
| 5.3.1 NESTED LOOPS .....              | 97  |
| 5.3.2 HASH JOIN .....                 | 97  |
| 5.3.3 SORT MERGE .....                | 98  |
| 5.4 Oracle 统计信息 .....                 | 98  |
| 5.4.1 默认统计信息 .....                    | 98  |
| 5.4.2 手动搜集统计信息 .....                  | 99  |
| 5.5 Oracle 直方图 .....                  | 102 |
| 5.5.1 直方图概要 .....                     | 102 |
| 5.5.2 直方图优化 .....                     | 103 |
| 5.6 Oracle 提示 .....                   | 107 |
| 5.6.1 Oracle 提示的语法及使用 .....           | 108 |
| 5.6.2 Oracle 提示失效场景 .....             | 108 |
| 5.7 Oracle SQL 跟踪 .....               | 109 |
| 5.7.1 10046 事件跟踪 .....                | 109 |
| 5.7.2 SQL 跟踪与 Tkprof .....            | 112 |
| 5.8 小结 .....                          | 113 |

## 第 6 章 Oracle CBO 优化 .....

|                             |     |
|-----------------------------|-----|
| 6.1 CBO 优化器模式与 CPU 成本 ..... | 114 |
|-----------------------------|-----|



## 第三篇 方法篇

|                                             |     |
|---------------------------------------------|-----|
| 6.1.1 CBO 优化器模式 .....                       | 114 |
| 6.1.2 CBO 优化器模式下的执行<br>计划调整 .....           | 115 |
| 6.1.3 CPU 成本 .....                          | 121 |
| 6.1.4 CPU 成本启用 .....                        | 124 |
| 6.2 谓词选择率与基数计算 .....                        | 124 |
| 6.2.1 单谓词选择率与基数计算 .....                     | 124 |
| 6.2.2 多谓词选择率与基数计算 .....                     | 129 |
| 6.3 表连接选择率与基数计算 .....                       | 132 |
| 6.3.1 表连接选择率与基数计算<br>解析 .....               | 132 |
| 6.3.2 表连接选择率与基数计算<br>验证 .....               | 134 |
| 6.4 Oracle 查询转换提示 .....                     | 136 |
| 6.4.1 dynamic_sampling 提示 .....             | 136 |
| 6.4.2 leading 与 ordered 提示 .....            | 138 |
| 6.4.3 index 提示 .....                        | 140 |
| 6.4.4 index_join 提示 .....                   | 141 |
| 6.4.5 index_ffs 提示 .....                    | 142 |
| 6.4.6 index_ss 提示 .....                     | 144 |
| 6.4.7 index_combine 提示 .....                | 145 |
| 6.4.8 use_concat 提示 .....                   | 147 |
| 6.4.9 expand 与 no_expand 提示 .....           | 148 |
| 6.4.10 merge 与 no_merge 提示 .....            | 148 |
| 6.4.11 unnest 与 no_unnest 提示 .....          | 150 |
| 6.4.12 push_pred 与 no_push_pred<br>提示 ..... | 151 |
| 6.4.13 push_subq 与 no_push_subq<br>提示 ..... | 152 |
| 6.4.14 pq_distribute 提示 .....               | 155 |
| 6.4.15 driving_site 提示 .....                | 158 |
| 6.5 小结 .....                                | 159 |

## 第 7 章 Oracle 备份恢复 .....

|                                     |     |
|-------------------------------------|-----|
| 7.1 备份恢复与日志记录体系 .....               | 162 |
| 7.1.1 Oracle 日志记录体系 .....           | 162 |
| 7.1.2 Oracle 备份恢复与日志记录<br>体系 .....  | 166 |
| 7.2 备份恢复与物理文件 .....                 | 183 |
| 7.2.1 备份恢复与控制文件 .....               | 184 |
| 7.2.2 备份恢复与数据文件（头） .....            | 201 |
| 7.2.3 备份恢复与日志文件（头） .....            | 204 |
| 7.3 备份恢复实现 .....                    | 205 |
| 7.3.1 Shutdown Clean 恢复 .....       | 206 |
| 7.3.2 Shutdown Abort（Crash）恢复 ..... | 210 |
| 7.3.3 冷备恢复 .....                    | 213 |
| 7.3.4 热备恢复 .....                    | 216 |
| 7.4 制定 RMAN 备份恢复计划 .....            | 224 |
| 7.4.1 RMAN 备份策略制定 .....             | 224 |
| 7.4.2 RMAN 备份脚本 .....               | 227 |
| 7.4.3 RMAN 日常备份脚本 .....             | 228 |
| 7.5 小结 .....                        | 235 |

## 第 8 章 Oracle 自动化运维管理

### 系统 .....

|                                 |     |
|---------------------------------|-----|
| 8.1 Oracle 自动化查询管理系统 .....      | 236 |
| 8.1.1 查询表空间使用情况 .....           | 237 |
| 8.1.2 查询 Lock 锁定信息 .....        | 239 |
| 8.1.3 查询事务运行状态 .....            | 240 |
| 8.1.4 查询 LibraryCache 命中率 ..... | 241 |
| 8.1.5 查询 BufferCache 命中率 .....  | 241 |
| 8.1.6 查询 Latch 命中率 .....        | 242 |
| 8.1.7 查询特定 Latch 命中率 .....      | 243 |



|        |                               |     |
|--------|-------------------------------|-----|
| 8.1.8  | 查询活动会话信息                      | 244 |
| 8.1.9  | 查询 SQL 执行情况                   | 245 |
| 8.1.10 | 查询 SQL 执行性能                   | 247 |
| 8.1.11 | 查询 SQL 执行计划                   | 248 |
| 8.1.12 | 查询游标使用情况                      | 249 |
| 8.1.13 | 查询日志切换频率                      | 251 |
| 8.1.14 | 查询会话等待情况                      | 252 |
| 8.1.15 | 查询系统等待情况                      | 253 |
| 8.1.16 | 查询系统平均等待                      | 255 |
| 8.2    | Oracle 自动化运维监控系统              | 257 |
| 8.2.1  | 自动化运维监控系统架构                   | 257 |
| 8.2.2  | 自动化运维监控系统环境<br>准备             | 258 |
| 8.2.3  | 部署操作系统级自动化运维<br>监控            | 260 |
| 8.2.4  | 部署数据库级自动化运维<br>监控             | 273 |
| 8.2.5  | 部署 Agent_All 与 Mail_All<br>调度 | 282 |
| 8.3    | 小结                            | 295 |

## 第 9 章 Oracle 自动化历史分析系统

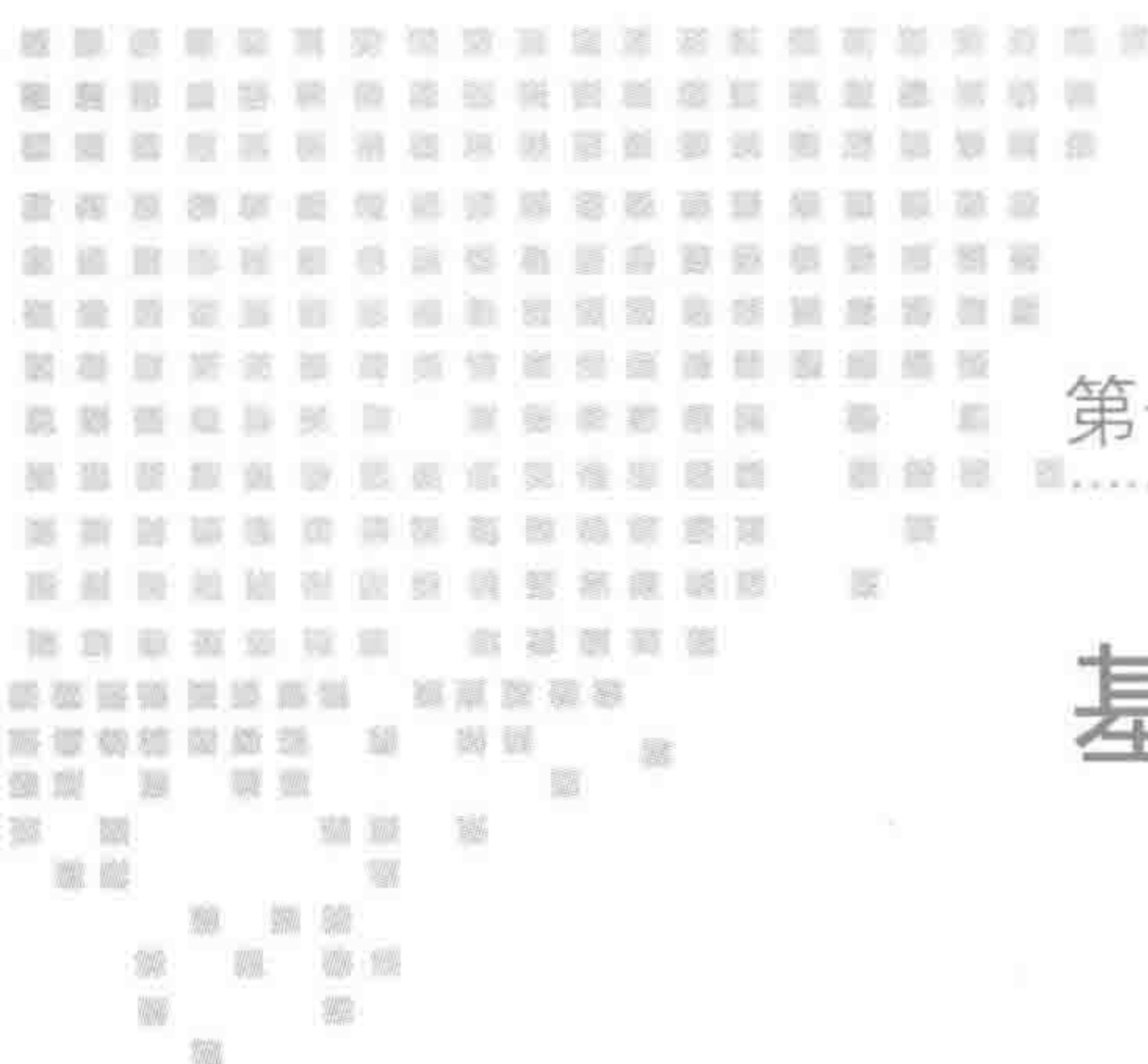
|       |             |     |
|-------|-------------|-----|
| 9.1   | 自动化历史分析系统   | 296 |
| 9.1.1 | 自动化历史分析系统架构 | 296 |

|       |                         |     |
|-------|-------------------------|-----|
| 9.1.2 | 自动化历史分析系统环境<br>准备       | 297 |
| 9.1.3 | 自动化历史分析系统部署             | 300 |
| 9.2   | 自动化历史分析系统与自动化<br>查询管理系统 | 320 |
| 9.2.1 | 历史分析系统接入查询管理<br>系统      | 320 |
| 9.2.2 | 查询管理系统中的历史分析            | 323 |
| 9.3   | 小结                      | 326 |

## 第 10 章 Oracle 自动化实时监控 系统

|        |                  |     |
|--------|------------------|-----|
| 10.1   | 自动化实时监控系统架构      | 327 |
| 10.2   | 自动化实时监控系统环境准备    | 328 |
| 10.2.1 | 服务器环境准备          | 328 |
| 10.2.2 | 数据库环境准备          | 334 |
| 10.3   | 自动化实时监控系统部署      | 349 |
| 10.3.1 | 部署实时监控系统主体<br>程序 | 349 |
| 10.3.2 | 部署实时监控系统监控项      | 364 |
| 10.4   | 自动化实时监控系统使用示例    | 378 |
| 10.4.1 | user_call 实时监控   | 378 |
| 10.4.2 | redo_gen 实时监控    | 379 |
| 10.5   | 小结               | 381 |





## 第一篇 *Part 1*

# 基础篇

- 第1章 Linux 下的 Oracle
- 第2章 Oracle 内存体系结构
- 第3章 Oracle 重做日志 (Redo)
- 第4章 Oracle 事务与回滚段





Chapter 1

第 1 章

# Linux 下的 Oracle

众所周知，Oracle 数据库与 Linux 操作系统是行业内使用最为广泛、功能最为强大的数据库 / 操作系统之一。随着企业业务的不断发展，越来越多的 Oracle 数据库被部署在 Linux 环境中，以提供核心业务数据的支撑。

与此同时，随着人们对 Oracle 运维管理经验的不断积累，大部分数据库管理员对 Linux 环境下的 Oracle 认识也越来越深刻，然而也有部分数据运维人员对于 Linux 环境下的 Oracle 内存分配存在疑惑。本章就 Oracle 与 Linux 内存关系进行讲解，帮助读者更为深入地理解 Linux 环境下的 Oracle 内存体系。

## 1.1 Linux 简介

我们知道，Unix 是一个功能强大、性能全面的多用户、多任务操作系统，可以应用到小至普通 PC，大至巨型计算机等多种平台上，是应用面最广、影响力最大的操作系统之一。

作为 Unix 的延续，Linux 是一种界面和性能与 Unix 相似甚至更好的操作系统，但 Linux 不源于任何 Unix，因此 Linux 不是 Unix，而是一个类似于 Unix 的产品。Linux 成功地模仿和延续了 Unix 的基本体系结构和设计风格，换言之，Linux 是一套兼容于 System V 以及 BSD Unix 的操作系统。对于 System V 来说，把程序源代码放到 Linux 重新编译之后就可以运行；而对于 BSD Unix 来说，它的可执行文件可以直接在 Linux 环境下运行。

通俗地讲，Linux 是一种遵从可移植操作系统环境标准规范（POSIX）的操作系统，它能够在普通 PC 上实现全部的 Unix 特性。Linux 受到广大计算机爱好者青睐的另一个原因



是：它继承了 Unix 的优秀功能，同时还弥补了 Unix 跨平台通用性的缺陷，使得 Linux 操作使用起来更为灵活、便捷。

## 1.2 Oracle 简介

早期是没有数据库这一说法的，这是因为在早期可以直接在内存中使用变量、数组之类的内存结构来存放数据，但是这些数据仅仅贮存在内存中，一旦内存释放，数据就会丢失。

后来人们觉得可以尝试使用平面文件（Flat File）的方式来永久存放数据，将数据记录在文件中，这样就可以在需要的时候将数据从平面文件中读取出来，这种数据存储模式被称为简单文件存储模式。

随着时代的发展，越来越多的文件需要存储在操作系统，导致从大量文件中检索特定的文件就变得非常困难，效率也极其低下。与此同时，人们对数据存储的需求日益增长，对数据存储的要求也越来越高，在这样的环境下，Oracle 应运而生。

作为市面上最优秀的数据库之一，Oracle 数据库系统是美国 Oracle 公司（甲骨文）提供的以分布式数据库为核心的软件产品，是目前最流行、应用最广泛的数据库之一。

Oracle 数据库主要特点有：

- ❑ 稳定性强；
- ❑ 可用性强；
- ❑ 可扩展性强；
- ❑ 数据安全性强。

当我们将 Oracle 部署在 Linux 平台后，就可以很好地发挥 Oracle 与 Linux 的功能特点：稳定、高效、灵活。接下来我们就 Oracle 与 Linux 内存体系进行介绍。

## 1.3 Linux 内存体系的优势

许多接触过 Oracle 的朋友可能会有一些感慨，在 Windows 操作系统和 Linux 操作系统下管理 Oracle 是完全不一样的。其实不尽然，Windows 下的 Oracle 在服务中只能看见一个类似 ORCL 的服务，同时在 Windows 任务管理器中也只能看见一个 Oracle 进程，而数据库后台进程则不可见。

Windows 下的 Oracle 服务，如图 1-1 所示。

|                              |        |      |
|------------------------------|--------|------|
| OracleJobSchedulerDEMON      | 禁用     | 本地系统 |
| OracleOra10g_homeTNSListener | 已启动 手动 | 本地系统 |
| OracleServiceDBMON           | 已启动 自动 | 本地系统 |

图 1-1 Windows 下的 Oracle 服务



Windows 任务管理器中的 Oracle 进程，如图 1-2 所示。

| 应用程序                | 进程              | 服务  | 性能        | 联网 | 用户 |
|---------------------|-----------------|-----|-----------|----|----|
| 映像名称                | 用户名             | CPU | 内存(专用工作集) |    |    |
| oracle.exe          | SYSTEM          | 00  | 492,280 K |    |    |
| java.exe            | yjfadmin        | 00  | 166,208 K |    |    |
| java.exe            | yjfadmin        | 00  | 165,240 K |    |    |
| explorer.exe        | yjfadmin        | 00  | 45,540 K  |    |    |
| svchost.exe         | SYSTEM          | 00  | 30,192 K  |    |    |
| httpd.exe *32       | SYSTEM          | 00  | 20,076 K  |    |    |
| BaiduSdSvc.exe *32  | SYSTEM          | 00  | 11,144 K  |    |    |
| popwndex.exe *32    | yjfadmin        | 00  | 10,356 K  |    |    |
| svchost.exe         | NETWORK SERVICE | 00  | 10,344 K  |    |    |
| lsass.exe           | SYSTEM          | 00  | 8,432 K   |    |    |
| mmc.exe             | yjfadmin        | 00  | 8,164 K   |    |    |
| svchost.exe         | SYSTEM          | 00  | 8,136 K   |    |    |
| BaiduSdTray.exe *32 | yjfadmin        | 00  | 7,832 K   |    |    |
| svchost.exe         | LOCAL SERVICE   | 00  | 7,640 K   |    |    |
| LogonUI.exe         | SYSTEM          | 00  | 6,604 K   |    |    |
| RavMonD.exe *32     | SYSTEM          | 00  | 5,964 K   |    |    |
| spoolsv.exe         | SYSTEM          | 00  | 5,812 K   |    |    |
| svchost.exe         | LOCAL SERVICE   | 00  | 5,552 K   |    |    |
| httpd.exe *32       | SYSTEM          | 00  | 5,020 K   |    |    |
| RsTray.exe *32      | yjfadmin        | 00  | 4,704 K   |    |    |
| services.exe        | SYSTEM          | 00  | 4,520 K   |    |    |
| taskmgr.exe         | yjfadmin        | 00  | 4,076 K   |    |    |
| svchost.exe         | LOCAL SERVICE   | 00  | 4,048 K   |    |    |
| svchost.exe         | NETWORK SERVICE | 00  | 4,044 K   |    |    |

图 1-2 Windows 任务管理中的 Oracle

**提示** Windows 环境下的 Oracle 后台进程以线程方式存在，因此在 Windows 环境无法详细查看后台进程信息，这对数据库故障排查带来诸多不便。

在 Linux 体系中，使用操作系统命令就可以精准地定位 Oracle 后台进程，还可以看见 Oracle 实例中内存段分配的信息等，为数据库的维护和故排提供了便捷的通道。

Linux 下的 Oracle，如下所示：

```
[oracle@oracle11g ~]$ ps -ef| grep oracle
avahi      2529      1  0 10:07 ?        00:00:00 avahi-daemon: running [oracle11g.local]
root       2634    2308  0 10:08 ?        00:00:00 sshd: oracle [priv]
oracle     2636    2634  0 10:08 ?        00:00:00 sshd: oracle@pts/0
oracle     2637    2636  0 10:08 pts/0    00:00:00 -bash
oracle     2673      1  0 10:10 ?        00:00:00 ora_pmon_oracle11g
oracle     2675      1  0 10:10 ?        00:00:00 ora_vktm_oracle11g
oracle     2679      1  0 10:10 ?        00:00:00 ora_gen0_oracle11g
oracle     2681      1  0 10:10 ?        00:00:00 ora_diag_oracle11g
oracle     2683      1  0 10:10 ?        00:00:00 ora_dbrm_oracle11g
oracle     2685      1  0 10:10 ?        00:00:00 ora_psp0_oracle11g
oracle     2687      1  0 10:10 ?        00:00:00 ora_dia0_oracle11g
oracle     2689      1  0 10:10 ?        00:00:00 ora_rman_oracle11g
oracle     2691      1  0 10:10 ?        00:00:00 ora_dbw0_oracle11g
oracle     2693      1  0 10:10 ?        00:00:00 ora_lgwr_oracle11g
oracle     2695      1  0 10:10 ?        00:00:00 ora_ckpt_oracle11g
oracle     2697      1  0 10:10 ?        00:00:00 ora_smon_oracle11g
oracle     2699      1  0 10:10 ?        00:00:00 ora_reco_oracle11g
oracle     2701      1  1 10:10 ?        00:00:01 ora_mmon_oracle11g
```



```

oracle 2703 1 0 10:10 ? 00:00:00 ora_mnnl_oracle11g
oracle 2705 1 0 10:10 ? 00:00:00 ora_d000_oracle11g
oracle 2707 1 0 10:10 ? 00:00:00 ora_s000_oracle11g
oracle 2715 1 0 10:10 ? 00:00:00 ora_p000_oracle11g
oracle 2717 1 0 10:10 ? 00:00:00 ora_p001_oracle11g
oracle 2719 1 0 10:10 ? 00:00:00 ora_qmnc_oracle11g
oracle 2734 1 1 10:10 ? 00:00:00 ora_cjq0_oracle11g
oracle 2736 1 0 10:10 ? 00:00:00 ora_q000_oracle11g
oracle 2738 1 0 10:10 ? 00:00:00 ora_q001_oracle11g
oracle 2742 1 6 10:10 ? 00:00:02 ora_j000_oracle11g
oracle 2760 2637 0 10:11 pts/0 00:00:00 ps -ef
oracle 2761 2637 0 10:11 pts/0 00:00:00 grep oracle

```

Linux 下的 Oracle 共享内存段，如下所示：

```
[oracle@oracle11g ~]$ ipcs -m
```

```

----- Shared Memory Segments -----
key          shmid      owner      perms      bytes      nattch     status
0x26adaee0 2326529    oracle     660        243269632  23

```

不难看出，相对于 Windows 来说，Linux 下的 Oracle 数据库管理更为便捷，数据库信息的获取也更为直观、有效。

## 1.4 Linux 内存体系与 Oracle 内存空间

### 1.4.1 Linux 用户空间与内核空间

一般来说，Linux 操作系统把虚拟地址空间划分为用户空间和内核空间。例如 x86 架构下的 32 位 Linux 虚拟地址空间是 4GB（0x0000 0000~0xffff ffff），其中大致将前 3GB（0x0000 0000~0xbfff ffff）划分为用户空间，后 1GB（0xc000 0000~0xffff ffff）划分为内核空间。

用户程序只能在用户模式（用户空间）下执行，而不能访问特权模式（内核空间）的数据，也不能跳转到内核代码执行，这样的设计可以保护内核，最主要的原因是当一个用户进程访问了非法地址，最坏的情况是该进程崩溃，而不会影响到内核和其他进程正常运行。

CPU 在产生中断或异常时会自动切换模式，由用户模式切换到特权模式，这时就可以允许跳转到内核代码中执行中断或异常服务程序。事实上，所有内核代码的执行都是从中断或异常服务程序开始的，整个 Linux 内核就是由各种中断处理和异常处理程序组成的。

Linux 下的 Oracle 内存分配如图 1-3、图 1-4 所示。



**提示** Oracle10g 中的 SGA 与 PGA 在 Linux 中是独立分配的，但是在 11g 中，默认情况下 SGA 内存区域包含了 PGA 的内存空间。

### 1.4.2 Linux 下的 Oracle 内存体系结构

我们知道，Oracle 主要是由内存结构和物理结构组成的，如图 1-5 所示。



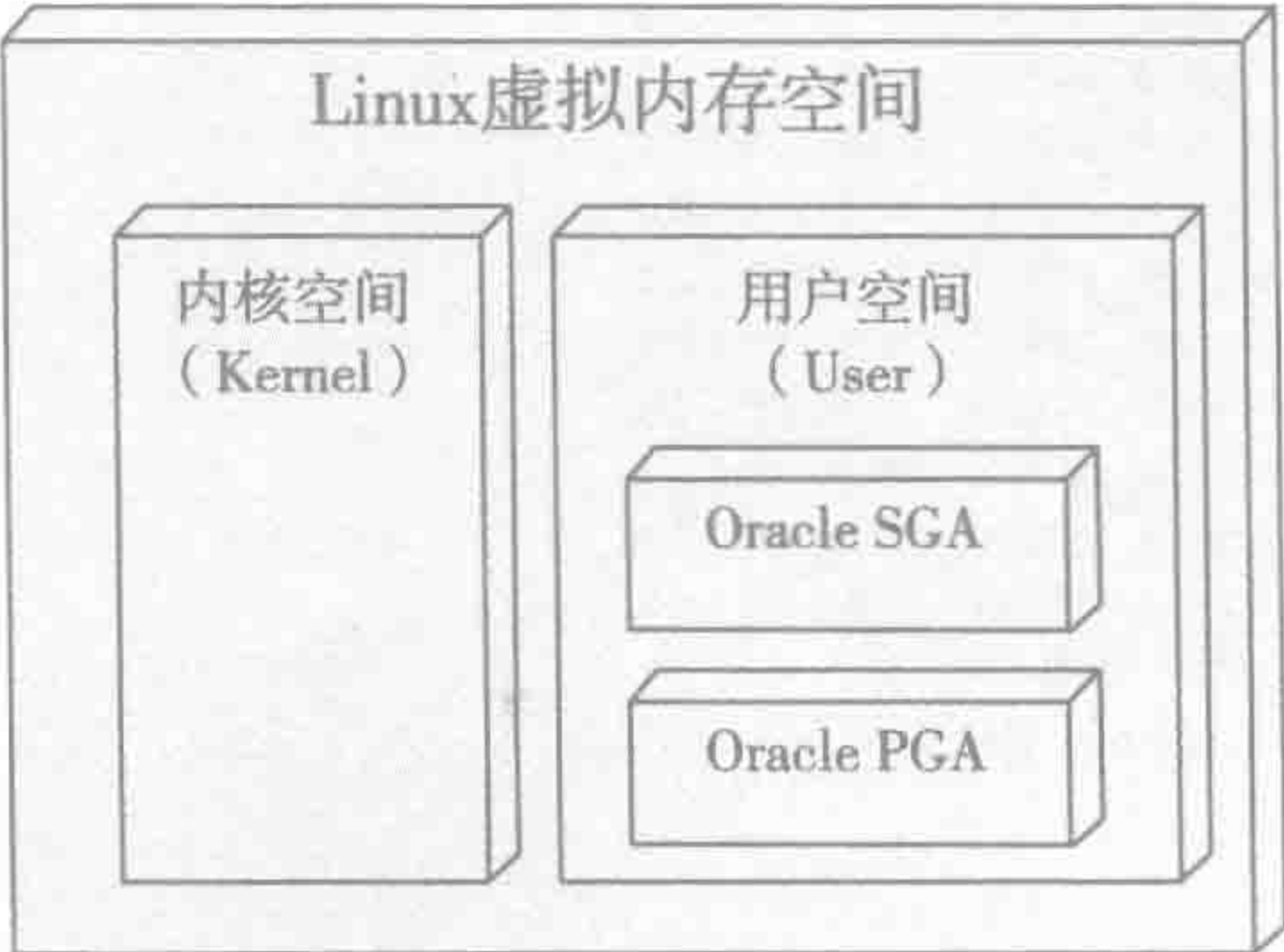


图 1-3 Linux 下的 Oracle 内存 (10g)

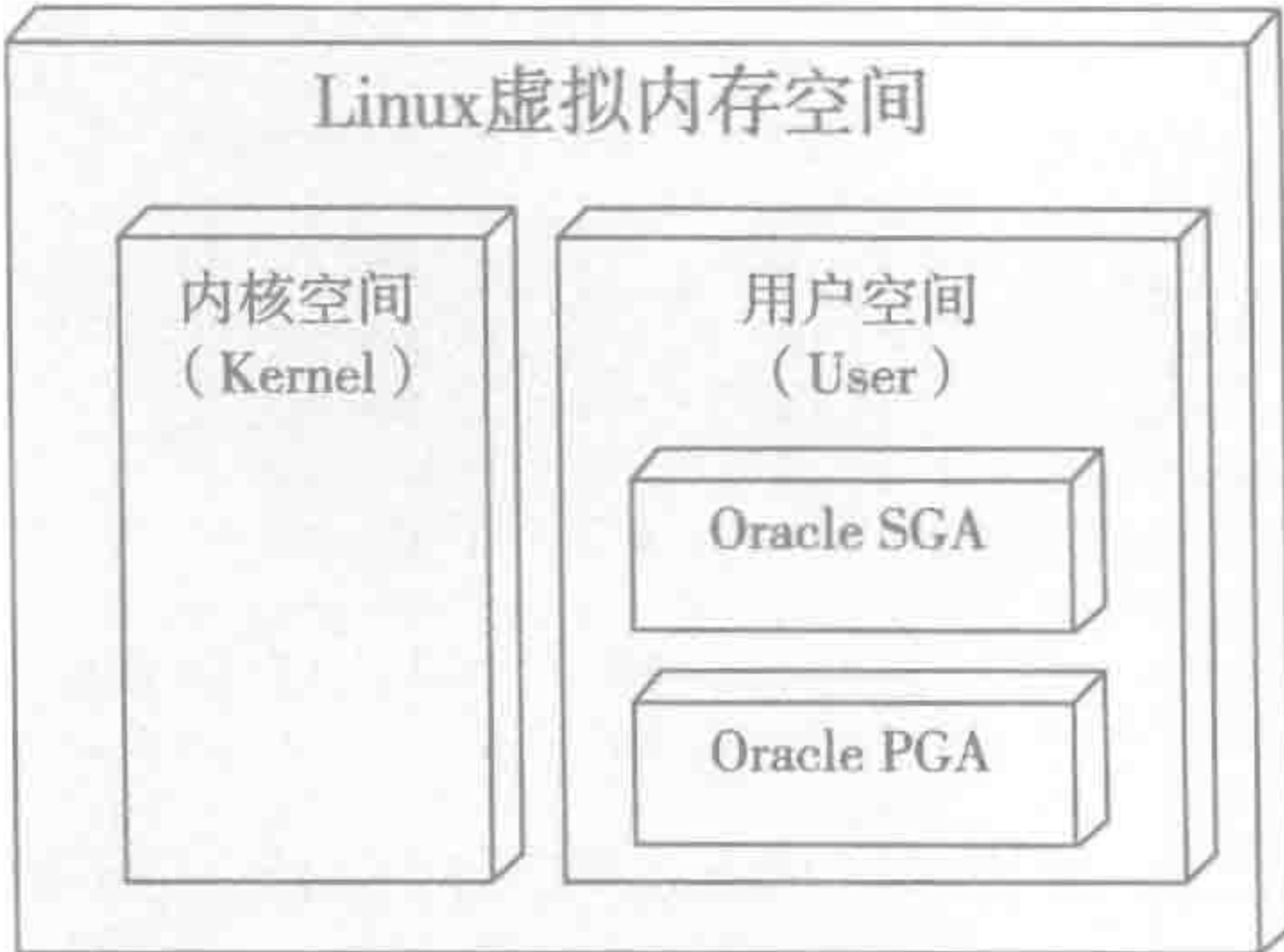


图 1-4 Linux 下的 Oracle 内存 (11g)

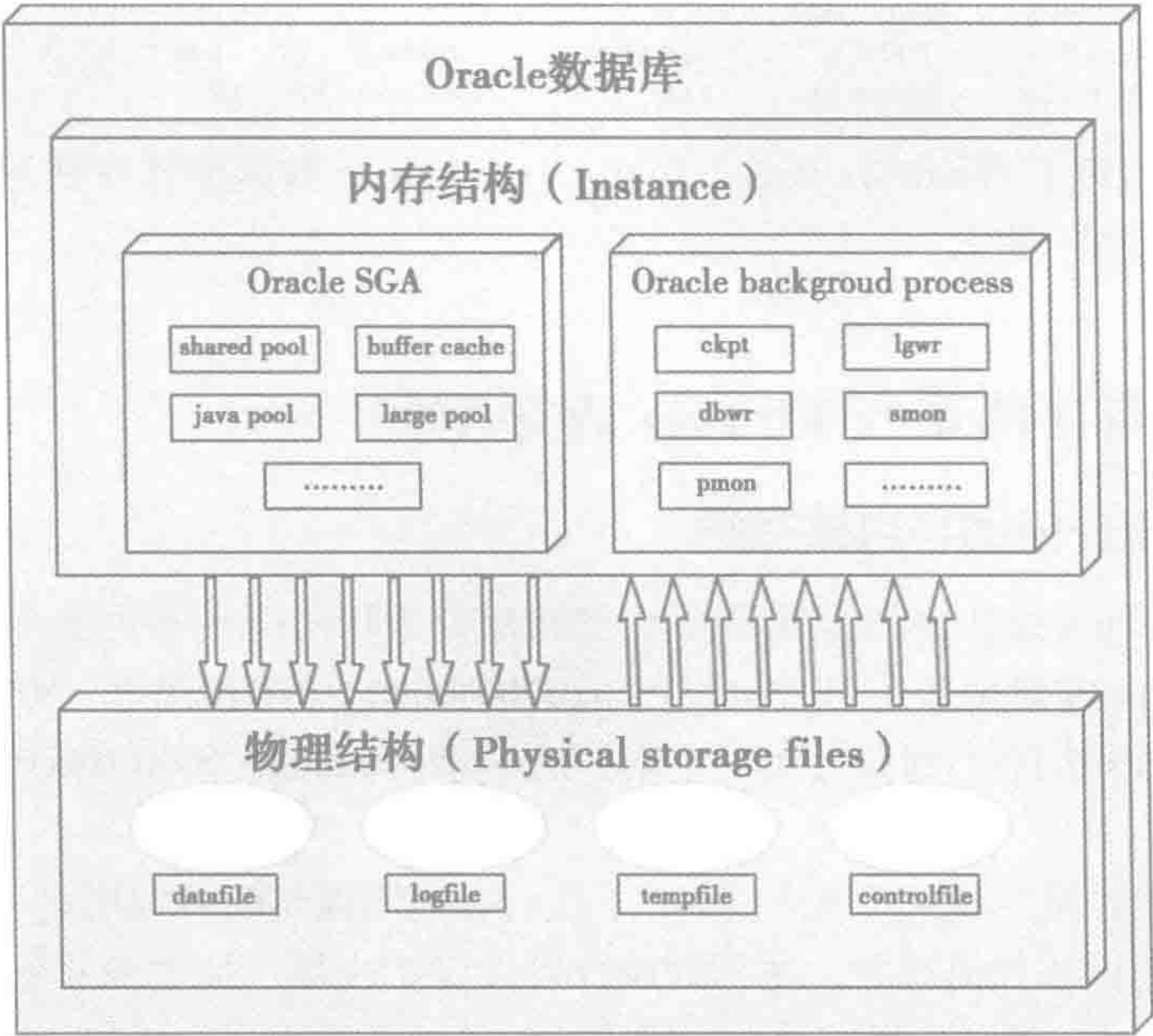


图 1-5 Oracle 体系结构

可以看出，Oracle 内存结构存在于操作系统的用户空间，因此我们在部署 Oracle 前，就必须事先在 Linux 操作系统层面配置好 Oracle 内存地址空间，这样才能成功地配置 SGA 与 PGA，从而避免在安装 Oracle 的时候出现类似“ORA-27102: out of memory”的错误。

1.4.3 Linux 下的 Oracle 内存分配

Oracle 内存结构处于 Linux 操作系统的用户空间，因此需要通过设置 Linux 操作系统内核参数来对 Oracle 内存进行分配。由于 Linux 延续了 SystemV IPC 以及后来的 POSIX IPC 通信标准，因此 Linux 下的内存分配主要由以下参数决定，如表 1-1 所示。



表 1-1 Linux 内存分配参数

| 内核参数名称 | 描 述                                                                          | 默 认 值   |
|--------|------------------------------------------------------------------------------|---------|
| SHMMAX | 单个共享内存段最大值 (字节)                                                              |         |
| SHMMIN | 单个共享内存段最小值 (字节)                                                              |         |
| SHMMNI | 系统共享内存段数量                                                                    | 4096    |
| SHMALL | 可用共享内存的总数量 (字节或者页面), 以字节为单位, 与 SHMMAX 一样; 以页为单位, 大小为 ceil (SHMMAX/PAGE_SIZE) | 2097152 |
| SHMSEG | 每进程最大共享内存段数量                                                                 |         |

 **提示** SHMALL 在很多系统上是用页面数而不是字节数来计算的。对于一般的 Linux 来说, Page 页的大小一般为 4096。

参数 SHMMAX 是 Oracle 最重要的 Linux 内核参数之一, 该参数以字节为单位限制单个共享内存段的最大值。与此同时, 参数 SHMALL 用于限制共享内存总数 (字节或者 Page 页)。对于 Oracle 而言, 必须保证这个数值足够大, 建议值为超过数据库共享内存与其他共享内存的总和。Oracle 官方文档对 Linux 参数限制描述如图 1-6 所示。

| Parameter           | Value                                       | File                                   |
|---------------------|---------------------------------------------|----------------------------------------|
| semmsl              | 250                                         | /proc/sys/kernel/sem                   |
| semmns              | 32000                                       |                                        |
| semopm              | 100                                         |                                        |
| semmni              | 128                                         |                                        |
| shmll               | 2097152                                     | /proc/sys/kernel/shmall                |
| shmmax              | Half the size of physical memory (in bytes) | /proc/sys/kernel/shmmax                |
| shmmni              | 4096                                        | /proc/sys/kernel/shmmni                |
| file-max            | 65536                                       | /proc/sys/fs/file-max                  |
| ip_local_port_range | Minimum:1024<br>Maximum: 65000              | /proc/sys/net/ipv4/ip_local_port_range |
| rmem_default        | 262144                                      | /proc/sys/net/core/rmem_default        |
| rmem_max            | 262144                                      | /proc/sys/net/core/rmem_max            |
| wmem_default        | 262144                                      | /proc/sys/net/core/wmem_default        |
| wmem_max            | 262144                                      | /proc/sys/net/core/wmem_max            |

图 1-6 Oracle 官方 Linux 内核参数

我们可以通过以下示例来验证 Linux 下的 Oracle (10g) 内存分配:

- 1) 查看 Oracle SGA 与 PGA 分配情况, 如图 1-7 所示。
- 2) 查看 Linux 系统共享内存分配情况, 如图 1-8 所示。

可以看到, Linux 系统分配了 3 个共享内存段, 每个共享内存段的大小为 2GB。

- 3) 查看 Linux 环境中的 sysctl.conf 配置, 如图 1-9 所示。

结合前面内容, 不难看出:

- ❑ 单个共享内存段最大值为 kernel.shmmax=2147483648B=2GB, 与图 1-10 一致;
- ❑ SGA=3 (共享内存段数量) × 2GB (单个内存段最大值) = 6GB;