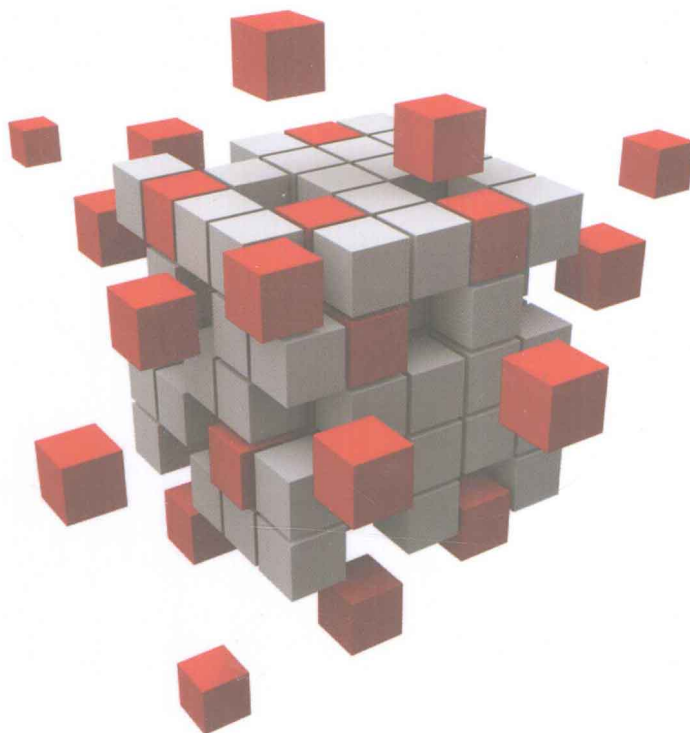


The Art of Oracle Database Performance Optimization

Oracle数据库 性能优化的艺术

刘华◎著



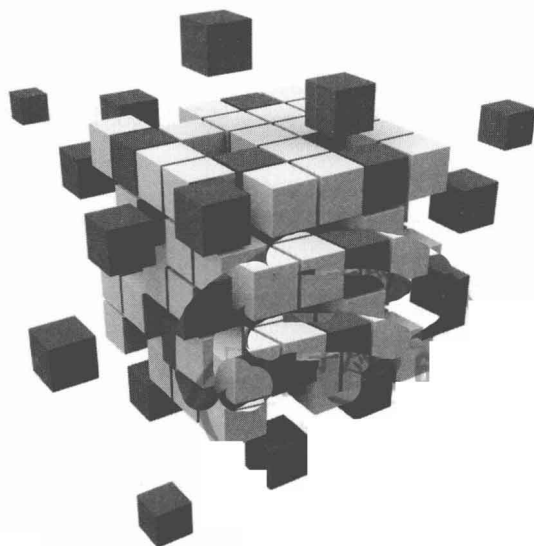
机械工业出版社
China Machine Press

数据库 技术丛书

The Art of Oracle Database Performance Optimization

Oracle数据库 性能优化的艺术

刘平◎著



机械工业出版社
China Machine Press

本书是资深数据库专家、UNIX 系统专家、系统架构师近 20 年工作经验的结晶。

本书内容高屋建瓴，用辩证法中的系统化分析方法，不仅从硬件（服务器系统、存储系统、网络系统等）、软件（操作系统、中间件系统、应用软件等）和应用场景（用户访问模式、用户使用频度、数据承载压力等）等多个相关联的维度深入阐述了具有普适意义的数据库性能评估与优化的思维方法和工作流程，而且还从流程的角度详细讲解了如何在数据库系统的架构阶段、设计阶段、开发阶段、部署阶段、运行阶段等各环节中去寻找性能问题的瓶颈和解决方案。

全书一共 12 章：第 1 章从系统工程的角度总结了性能问题可以分为哪些阶段去考虑、性能问题有哪些种类，以及性能改进的工程方法等；第 2 章详细讲解了在数据库的设计阶段如何去考虑数据库的性能问题；第 3~6 章从物理数据库部署、数据库访问设置、硬件资源设定、存储空间和内存资源的使用、实例配置与缓存优化等多方面讲解了数据库系统在部署阶段的性能优化问题；第 7 章探讨了如何平衡数据库可靠性和性能之间的矛盾；第 8 章完整呈现了 Oracle 数据库的表分区能力，是解决与大规模数据处理相关的性能问题的关键；第 9 章从索引和执行计划的角度讲解了数据库性能优化的方法；第 10~12 章着重介绍了在数据库系统部署并运行后，如何获得数据库运行时的性能状态，如何对正在使用的各种资源进行分析，分析时使用什么工具，以及如何解读这些工具返回的信息。

本书重在“授人以渔”，虽然主要内容是围绕 Oracle 数据库系统展开的，但是书中的观点同样适用于 DB2、SQL Server、MySQL、PostgreSQL 等数据库系统。

封底无防伪标均为盗版

版权所有，侵权必究

本书法律顾问 北京市展达律师事务所

图书在版编目（CIP）数据

Oracle 数据库性能优化的艺术 / 文平著. —北京：机械工业出版社，2012.6

ISBN 978-7-111-38859-3

I. O… II. 文… III. 关系数据库—数据库管理系统 IV. TP311.138

中国版本图书馆 CIP 数据核字（2012）第 131097 号

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：李 荣

冀城市京瑞印刷有限公司印刷

2012 年 7 月第 1 版第 1 次印刷

186mm×240mm·25.25 印张

标准书号：ISBN 978-7-111-38859-3

定价：79.00 元

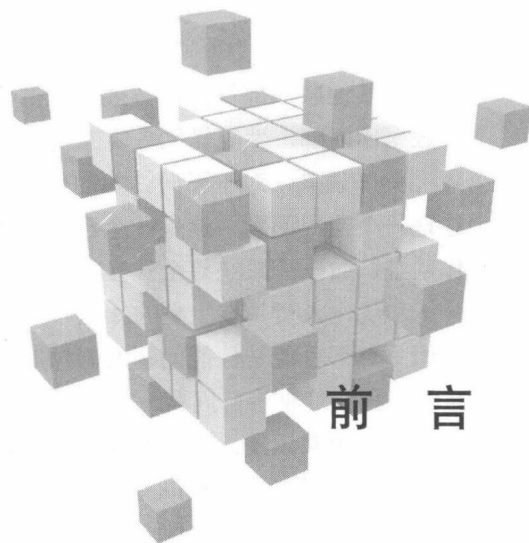
凡购本书，如有缺页、倒页、脱页，由本社发行部调换

客服热线：(010) 88378991；88361066

购书热线：(010) 68326294；88379649；68995259

投稿热线：(010) 88379604

读者信箱：hzjsj@hzbook.com



前言

数据库技术在 IT 系统中的广泛应用已不少于二十年。在这二十几年中，我们看到了数据库技术的起承与转合，产品的此消和彼长，供应商的发展或萎缩。太多的变化使我们有些茫然，快速的更迭使我们疲于奔命，作为数据库从业者，我们在技术方向上应该如何跟随？我们的行动方法应该如何适应？

这是一个大问题！

辩证法中很朴素的两条准则也许能帮到我们：第一，区分现象和本质；第二，要用系统的方法论来分析问题。

这点对于数据库优化工程来说更为重要！为什么这么说呢？

随着这二十年的 IT 应用持续深入，数据库的应用已经进入了相对成熟的时代。作为 IT 基础设施的核心组成部件，数据库早已不是可以孤立存在的系统——它与服务器系统、存储系统、网络系统等硬件紧密相连，它与操作系统、中间件系统、应用编码等软件紧密相连，它与用户访问模式、用户使用频度、数据承载压力紧密相连。**这种多重关联特性决定了数据库性能的评估与优化就是一门综合的技术与艺术，是一项相当严密的系统工程。**在这个系统工程中，我们需要透过各种表象进行全系统评估从而了解实质，并在最后得到相对精确和有效的结论与措施。

那么，能不能找到一种“具有普世意义的药方”呢？

很遗憾，我认为这种想法没有任何存在的可能性！IT 应用的显著特点就是需求个性化、部署个性化、使用个性化。在这个高度个性化的环境中，提炼出一个普遍适用的数据库性能优化准则肯定是不可行的。

但这并不是说没有规律可循，或者说，“具有普世意义的流程”是存在的，这个流程可以包括如下的关键环节——从什么样的角度切入性能问题，如何保证数据库运行效率与 IT

系统中其他联动部件保持协调，数据库性能优化的工程过程要有哪些阶段。该流程并未局限于数据库内部，性能优化是要围绕 IT 应用整体来展开的。

具体说来，对于面向性能的优化，大家需要有四个共识。第一个共识是，要充分意识到这是一个全局问题，这是一个应用程序、部署方式、存储设备、服务器设备、网络拓扑、操作系统、数据库系统等全员参与的优化问题。由于很多问题是在数据库以外的，这要求性能分析和优化人员具有完整的知识体系——这是一个巨大的挑战。但是没有办法，IT 系统中的任何一点都可能是性能问题的麻烦制造者。在这点上，局部的观点要服从全局的观点——从系统工程角度来看待问题。

第二个共识是，需要在设计阶段讨论性能问题，并及时做好性能风险上的应对策略。这里的设计不仅仅指应用的架构设计，还特指应用中的数据库逻辑结构与物理结构部署。数据库设计中的需求实现以及设计中的优化考虑，是数据库性能优化的基本决定要素。如果不能在设计上考虑优化要素，那么这样的架构设计就是先天不足的架构设计。

第三个共识是，需要在开发阶段讨论性能问题。很多工程案例告诉我们，应用的编码构成决定了应用在性能方面的主要特征。同样的应用逻辑，不同的代码实现，其效率可能是不同的。编写性能可以达到最佳同时系统开销最小的应用模块，是编码中事关性能的主要任务。但遗憾的是，在大多数系统中，由于编程人员的技术素质、项目完成工期上的压力、项目预算上的限制等缘由并未做完整的优化处理。

第四个共识是，在数据库性能优化工作中，如果系统处于运行阶段，这时优化人员的手段是极其有限的，优化效果和作用也相对具有局限性。从优化工作的贡献率来说，数据库调优成本会随着 IT 应用的生命周期进程而增加，但调优收益却是同步降低并逐渐趋于零的。

阅读本书之前，我们需要建立如上的几个共识，因为本书就是按照这样的共识和这样的顺序完成写作和表述的。

本书主要介绍的是 Oracle 数据库应用系统的优化问题，但是本书的技术观点不限于此，对于其他数据库（DB2 也好，SQL Server 也好，或者是 MySQL、PostgreSQL 等中小规模数据库也好）来说，本书的技术观点同样适用。

本书从前至后分为若干部分来描述数据库性能系统工程中的若干环节。其中的第一个环节便是数据库设计问题。把该问题作为本书的首要描述内容，就是因为最优的性能来源于最恰当的数据库设计方案。在工程实践中，很多数据库设计方案是由不当设计造成的，所以，在设计阶段，我们必须要考虑数据库设计问题。这里面包括逻辑数据库设计、物理数据库设计、物理数据库部署问题，分别就是本书第 1~3 章的内容。

正确的数据库物理部署和适应性配置策略，恰当的数据库访问设置和硬件资源的使用设定，是数据库优化运行的一个重要性能因素。这里主要涉及物理资源在操作系统层面的计算资源分配，存储空间使用的设置，内存资源的使用等问题。这些分别就是本书第 4~6 章的内容。

数据库可靠性和数据库性能往往会产生矛盾。那么，在 Oracle 中，我们如何考虑这个矛盾结合体呢？这是本书第 7 章试图要说明的内容。

对于 Oracle 来说，Oracle 也许更适用于承载大规模数据，因为 Oracle 有着其他数据库平台所少有的大规模数据处理解决方案——多分区技术。本书第 8 章将向读者完整呈现 Oracle 数据库的表分区能力。

数据库设计中的一个重要内容是查询设计与索引配套。Oracle 在处理查询时需要考量 SQL 查询的内容和对应数据库中表与索引的存在匹配关系，并因此由数据库优化器提交 SQL 执行计划。Oracle 支持各种索引类型，那么我们如何使用这些索引类型，在什么情况下使用什么索引呢？这是本书第 9 章所试图描述的。

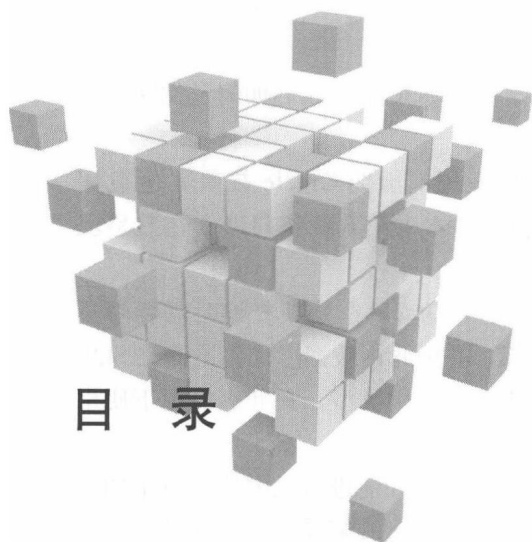
在数据库系统部署运行后，如何获得数据库运行性能状态，如何进行数据库运行中的资源使用分析，在分析时使用什么样的工具，以及这些工具返回信息的解读方法，这是本书第 10~12 章着重介绍的内容。这部分面向系统的运行监测和运维，试图解决我们在系统上线后“该做什么”的问题。

本书虽然是面向性能问题的，但是我也自知并不能解决所有问题。而且，本书中提到的技术观点和做法，可能在一些具体场景下适用，在其他一些场景下却并不适用，这点希望大家能够注意到。同时，作为本书作者，我在本书中没有写入任何所谓的“绝招、法宝”。近二十年从事数据库技术研究和实践的成功经验与失败教训告诉我，从来就不曾有绝招和法宝！能相信和依赖的，只有我们自己正确的系统观和工程实践上的科学精神，以及个人的勤奋和认真罢了！

恰在本书写作完成后，中国的 IT 业发生了一次不小的“地震”——华为正式进入 IT 领域，并成立 IT 产品线。鉴于对华为精神的敬仰，我接受了华为工作聘请，任职为华为 IT 架构师，进行数据库领域的产品架构设计和技术研发。我希望我的知识、经验，我的工作热情、潜能，能为华为在 IT 领域的拓展贡献自己的光和热。

同时，也祝本书的各位读者工作顺利、事业有成！

文平



前言

第 1 章 综述：优化是一个系统工程 / 1

- 1.1 性能问题是一个系统工程 / 3
 - 1.1.1 考虑性能问题的不同阶段 / 3
 - 1.1.2 考虑性能问题的不同方面 / 4
 - 1.1.3 性能架构中的问题分类 / 5
- 1.2 性能改进的工程方法 / 23
 - 1.2.1 不要希望一步到位！ / 23
 - 1.2.2 有时也能一步到位？ / 24
 - 1.2.3 性能改进工作流程 / 26
 - 1.2.4 建立性能分析流程图 / 28
 - 1.2.5 编程中应尽量避免的问题 / 30
 - 1.2.6 紧急事件性能处理 / 35

第 2 章 设计与优化：物理数据库结构设计 / 39

- 2.1 数据库的特定运行平台 / 40
- 2.2 物理设计与性能的关系 / 41
- 2.3 应用框架与性能特征 / 43
- 2.4 物理表设计的关注点 / 44
 - 2.4.1 从逻辑模型到物理模型 / 44
 - 2.4.2 物理数据库设计的范围 / 48

- 2.4.3 规范化的设计 / 49
- 2.4.4 反规范化设计 / 55
- 2.4.5 另类反规范化 / 59
- 2.4.6 是否反规范化 / 59
- 2.5 索引的初始设计 / 60
 - 2.5.1 索引对象的基本结构 / 61
 - 2.5.2 索引的可选择性指标 / 62
 - 2.5.3 索引的双面性特征 / 62
 - 2.5.4 索引的类型 / 64
 - 2.5.5 索引的初始确定 / 69
 - 2.5.6 临时表的使用 / 71

第 3 章 部署与优化：准备基本运行环境 / 73

- 3.1 性能问题在表象上的欺骗性 / 74
- 3.2 理解数据库需要的系统资源 / 76
 - 3.2.1 I/O 资源的使用 / 78
 - 3.2.2 内存资源的使用 / 82
 - 3.2.3 CPU 资源的使用 / 86
 - 3.2.4 解读内存与 CPU 状态 / 93
- 3.3 小型机数据库部署准备 / 94
 - 3.3.1 AIX 的部署准备 / 94
 - 3.3.2 HP-UX 的部署准备 / 98
 - 3.3.3 Solaris 的部署准备 / 102
- 3.4 X86 系统数据库部署准备 / 107
 - 3.4.1 Linux 系统的部署准备 / 107
 - 3.4.2 Windows 系统的部署准备 / 110

第 4 章 部署与优化：数据库的优化部署 / 112

- 4.1 实例初始配置注意点 / 114
- 4.2 数据库空间使用关注 / 119
- 4.3 表空间的创建管理 / 130
- 4.4 创建表时的性能因素 / 135
- 4.5 数据导入与索引维护 / 144

第 5 章 实例优化：配置高效运行环境 / 145

- 5.1 实例配置与缓存优化 / 146
 - 5.1.1 启用自动内存管理 / 148
 - 5.1.2 主机内存分配原则 / 152
 - 5.1.3 使用缓存顾问工具 / 155
 - 5.1.4 缓存的命中率问题 / 158
 - 5.1.5 缓存命中率分析 / 160
 - 5.1.6 精细化缓存的配置 / 161
- 5.2 SQL 与缓存的使用 / 165
 - 5.2.1 缓存优化 / 165
 - 5.2.2 查看数据缓存内容 / 165
 - 5.2.3 共享池的效率检查 / 166
 - 5.2.4 重做日志缓存问题 / 173
- 5.3 PGA 内存分配原则 / 175
- 5.4 Oracle 虚拟化关注点 / 176
 - 5.4.1 I/O 问题上的关注点 / 178
 - 5.4.2 内存问题上的关注点 / 178
 - 5.4.3 CPU 问题上的关注点 / 179
 - 5.4.4 故障恢复问题的解决 / 179

第 6 章 存储优化：配置高效存储环境 / 181

- 6.1 I/O 问题的表象与内涵 / 182
- 6.2 存储介质的基础配置 / 183
 - 6.2.1 使用逻辑卷技术 / 184
 - 6.2.2 设定存储参数 / 187
- 6.3 存储 I/O 能力探究 / 197
 - 6.3.1 Orion 的安装 / 197
 - 6.3.2 Orion 的使用 / 198

第 7 章 综合部署：Oracle 可靠性设置 / 201

- 7.1 数据库设置的综合考虑 / 202
 - 7.1.1 设置数据库的内存使用 / 203

- 7.1.2 数据库的进程使用设置 / 208
- 7.1.3 设置数据库 I/O 模式 / 211
- 7.1.4 设置数据库的归档模式 / 213
- 7.1.5 使用闪回恢复区归档 / 214
- 7.1.6 设置冗余控制文件 / 215
- 7.1.7 保留控制文件信息 / 216
- 7.1.8 设置联机重做日志文件 / 217
- 7.1.9 设置联机重做日志组数 / 218
- 7.1.10 设置数据块校验和检查 / 220
- 7.1.11 打开性能时间统计 / 221
- 7.1.12 表空间和段管理方式 / 221
- 7.1.13 设定多临时表空间和组 / 222
- 7.1.14 使用自动撤销管理 / 223
- 7.1.15 设置空间错误延时 / 223
- 7.1.16 设置数据库闪回 / 224
- 7.2 RAC 下的设置 / 225
 - 7.2.1 注册远程监听器的实例 / 226
 - 7.2.2 设置内部互连网络 / 226
- 7.3 加密表数据 / 227
- 7.4 加密表空间 / 231

第 8 章 对象优化：大表分区方式部署 / 232

- 8.1 分区技术适用的范围 / 233
- 8.2 分区技术的基本概念 / 234
 - 8.2.1 分区表的结构 / 234
 - 8.2.2 分区部署的收益 / 235
 - 8.2.3 分区部署的成本 / 236
 - 8.2.4 索引分区 / 237
- 8.3 分区表的管理操作 / 238
 - 8.3.1 面向分区的管理 / 238
 - 8.3.2 面向分区的操作 / 238
 - 8.3.3 面向分区的恢复 / 239
- 8.4 分区的技术实现 / 239

- 8.4.1 基础分区的实现 / 240
- 8.4.2 组合分区的扩展 / 246
- 8.4.3 扩展分区技术 / 249
- 8.4.4 分区表的数据压缩 / 254
- 8.4.5 查看数据存储位置 / 255
- 8.4.6 分区键数据的改变 / 256
- 8.4.7 分区方式选择问题 / 257
- 8.4.8 使用分区顾问工具 / 258

第 9 章 索引定义：索引设置与执行计划 / 260

- 9.1 索引对象 / 261
- 9.2 Oracle 优化器 / 263
 - 9.2.1 优化器的选择 / 264
 - 9.2.2 优化器的设置 / 266
 - 9.2.3 数据扫描问题 / 268
- 9.3 索引的技术指标 / 285
 - 9.3.1 索引的选择性 / 285
 - 9.3.2 索引的簇化率 / 286
 - 9.3.3 索引二元高度 / 289
 - 9.3.4 直方图的使用 / 290
- 9.4 索引与 SQL 优化 / 294
- 9.5 避免索引不作为 / 296
 - 9.5.1 非等式运算 / 297
 - 9.5.2 Null 值的比对 / 298
 - 9.5.3 函数的调用 / 301
- 9.6 创建虚拟的索引 / 303
- 9.7 创建压缩的索引 / 305
- 9.8 索引的使用监测 / 305
 - 9.8.1 创建不可视索引 / 305
 - 9.8.2 索引的可用状态 / 308
 - 9.8.3 监测索引的使用 / 309
- 9.9 对分区表建立索引 / 310

第 10 章 实例效率：识别数据库内部作为 / 312

- 10.1 研究数据库运行效率 / 313
 - 10.1.1 定义和量化性能 / 314
 - 10.1.2 统计分析的要素 / 314
 - 10.1.3 性能状态的统计 / 327
 - 10.1.4 其他相关统计 / 334
- 10.2 操作系统的使用方法 / 338
 - 10.2.1 性能监视的手段 / 338
 - 10.2.2 使用 vmstat / 338
 - 10.2.3 使用 iostat / 341
 - 10.2.4 使用 sar / 345
 - 10.2.5 使用 nmon 命令 / 347
- 10.3 调整计划探讨 / 349

第 11 章 优化践行：数据库层面的优化 / 351

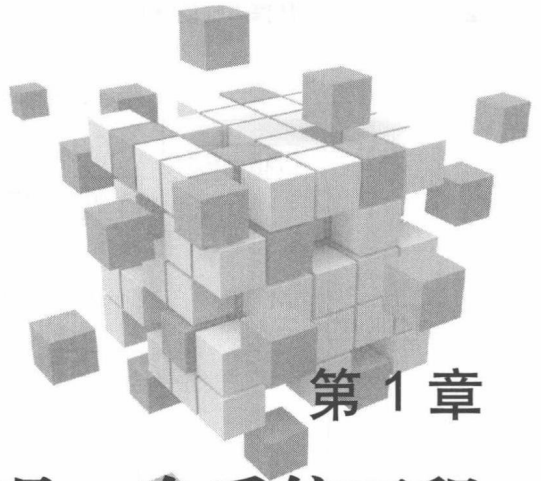
- 11.1 Oracle 自动统计 / 352
 - 11.1.1 统计数据的解释 / 354
 - 11.1.2 汇总的统计数据 / 355
- 11.2 使用 AWR / 355
 - 11.2.1 在 OEM 中访问 AWR / 357
 - 11.2.2 用 API 管理 AWR 快照 / 359
 - 11.2.3 使用度量基线 / 360
 - 11.2.4 AWR 工作报告 / 362
- 11.3 使用 ADDM / 363
 - 11.3.1 ADDM 内部结构 / 364
 - 11.3.2 ADDM 使用示例 / 365
 - 11.3.3 ADDM 环境设置 / 367
 - 11.3.4 ADDM 手工获得 / 367

第 12 章 优化践行：操作系统层面的优化 / 369

- 12.1 平台的性能相关性 / 370
- 12.2 分配和调度内存资源 / 372

- 12.2.1 使用大的内存页面 / 373
- 12.2.2 设定内存替换阈值 / 375
- 12.2.3 设定内存空闲阈值 / 379
- 12.2.4 对应设置应用内存 / 380
- 12.3 设置系统进程资源 / 381
- 12.4 系统 I/O 资源的使用 / 382
 - 12.4.1 优化使用异步 I/O / 382
 - 12.4.2 优化文件系统挂载 / 384
 - 12.4.3 优化应用 I/O 模式 / 386
- 12.5 关闭系统中无用的服务 / 387
- 12.6 监控主机资源使用 / 387
 - 12.6.1 服务器性能监控 / 388
 - 12.6.2 分析系统的性能 / 389

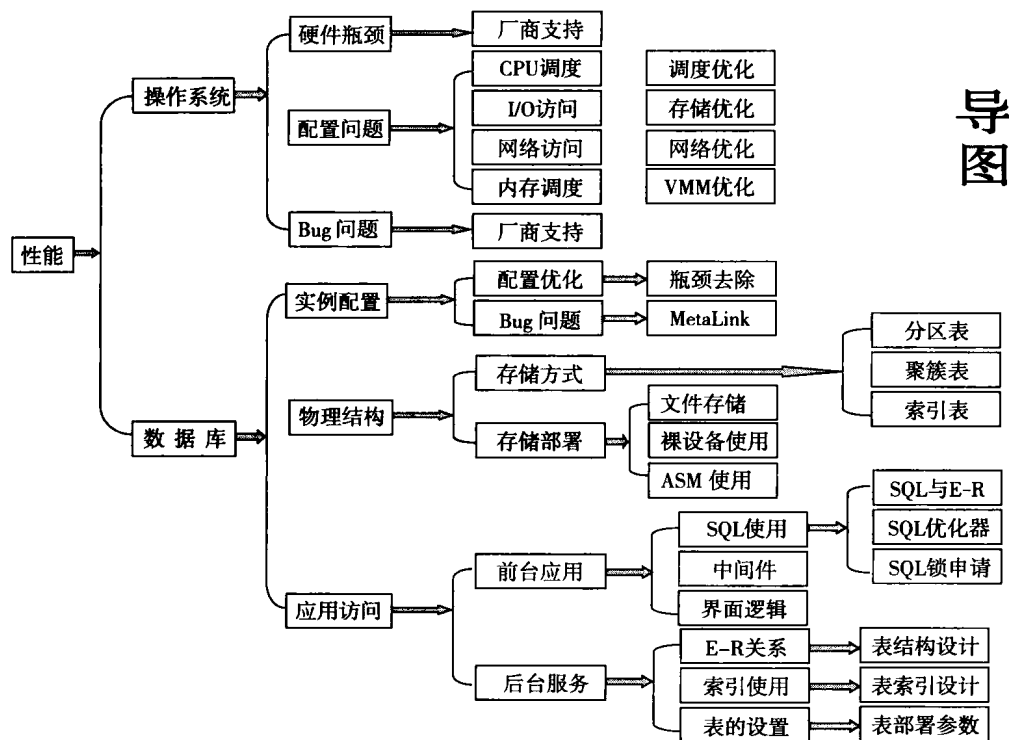
写在最后 / 391



第 1 章

综述：优化是一个系统工程

- 1.1 性能问题是一个系统工程
- 1.2 性能改进的工程方法



面向数据库系统性能的系统工程

本章将着力于阐述本书所倡导的一个基本技术观点——性能优化是要在全系统范围内考虑的一个系统工程。

众所周知，性能是涉及范围广泛的一个综合性问题。面对性能问题，有时候用户看到的仅仅是一个表象：“慢”！事实上，这个词太过含糊，其背后所涉及的问题可能包括：

- ❑ 哪台机器“慢”？
- ❑ 哪个业务“慢”？
- ❑ 什么时候“慢”？
- ❑ 什么操作“慢”？
- ❑ 谁操作时“慢”？

.....

那么，我们就来尝试解决这个含糊的“慢”的问题。现在站到服务器边上，看看机架上纵横交错的连线，还有那如繁星般闪烁的群灯，然后问自己：“这个家伙到底哪里出了问题呢？”也许我们会考虑到如下一些方面：

- ❑ 可能是 CPU 过载了？一定有这可能！
- ❑ 是内存调度颠簸？一定有这可能！

- 网络传输出现瓶颈？一定有这可能！
- 磁盘系统访问延迟严重？一定有这可能！
- I/O 通道有硬件问题？一定有这可能！
- 内存泄露资源耗尽？一定有这可能！
- 系统出现 Bug？一定有这可能！
- 业务数据量太大？一定有这可能！
- 用户连接太多了？一定有这可能！
- 用户正在执行某报表业务？一定有这可能！
- 用户正在执行备份？一定有这可能！
- 业务系统模块改变？一定有这可能！
- 中间件系统的问题？一定有这可能！
- 某些我们“尚不知道”的可能？一定有这可能！

对于上述的诸多可能，如果要尽述其细节，可能会列出数百种，甚至更多！假如我们能够列出一份排除清单，并逐一排除，相信很快就会发现一个基本的事实：我们已陷入了一个“问题”的海洋，并且难以自拔。

我写上述文字的目的不是想制造一种技术上的白色恐怖氛围，而是想说明一个技术观点：没有行之有效的方法，不掌握解决问题的规律，那么事倍功半是必然的！

也就是说，技术实现是有其规律的，这里冒昧地将科学泰斗钱学森的“系统工程”理论引用到性能优化这样的工程实践中来，全面系统地看待问题，找到方向，细粒度地解决问题，获得工程收益。

1.1 性能问题是一个系统工程

如前所述，性能问题是一个系统工程。在这个工程中，不同的时间段具有不同的工作方向和工作内容，不同的技术方向又有不同的问题考量和实施纲要。在这里，我们将考虑它的不同阶段和各方面的问题。

1.1.1 考虑性能问题的不同阶段

在实际操作中，我们会面临几种性能相关问题的工程场景：

- 系统未开发，尚处于系统架构之初。
- 系统开发中，已进入实质的设计阶段。
- 系统上线运行，已进入综合调试阶段。
- 系统上线后，已进入运维等日常管理阶段。

站在不同的工程场景中，我们在性能方面可以完成的工作也会相应有所不同。毫无疑问，最理想的系统性能设计要从应用设计阶段开始，并在整个系统开发周期中实现。待系统进入生产环境后还会继续调整，动态优化系统性能。

在初始的系统架构设计阶段，也许我们着重考虑的优化任务应包括以下内容：

- 服务器硬件上的选型。
- 系统可扩展性的设计。
- 系统容灾可行性的考虑。

当系统进入开发周期，并进入实质的设计阶段时，我们应该考虑以下内容：

- 系统的后台结构设计。
- 应用的前台模块与代码设计。
- 用户访问中的并发问题。
- 应用访问中的工作载荷要素。

当系统上线并经过测试开始运行时，我们可以完成的优化任务包含以下内容：

- 操作系统、数据库系统部署的合理性再研究。
- 后台数据库部署中的微观调整与实施。
- 部分应用代码的性能相关重构与库结构微调。
- 后台系统运维中的硬性任务安排设置。
- 应用部署结构的渐进性再调整。

待应用系统上线正式运行，已进入运维等日常管理阶段后，虽然我们可做的工作非常有限，但还是可以在以下事情上有所作为：

- 恰当组织数据存储结构，以优化性能。
- 恰当安排应用任务的执行区间，以优化性能。
- 恰当设置系统容灾结构，以避免对整体性能造成影响。
- 微观调整数据库对象（索引、物化视图等）的设计，以优化 SQL。

不同的场景下，我们可以设置不同的性能优化方向，包括从宏观到微观，从整体到局部。

1.1.2 考虑性能问题的不同方面

图 1-1 是我在性能优化工作中考虑问题所使用的综合导图。

假设图 1-1 所示的内容是正确且全面的（图 1-1 仅表述了我本人的技术观点，在一定程度上，作者认为该图是不全面甚至有技术缺陷的）。从图中可以看到，性能与“数据库服务器的方方面面（从硬件到系统、从平台到应用、从访问量到工作载荷等）均相关。

这就是解决性能问题很“困难”的根源所在。