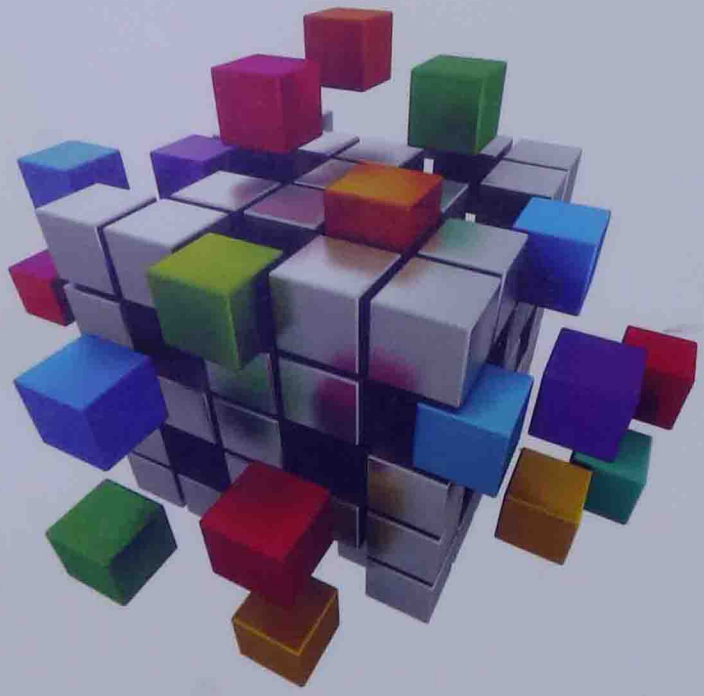
拥有10余年数据库从业经验的资深数据库架构师撰写，盖国强等多位业内数据库专家联袂推荐
秉承大道至简思想，从内部扩展、横向扩展和纵向扩展3个维度对架构与设计高并发Oracle数据库系统的思想、方法、核心技术进行深入讲解和剖析
国内首部讲解内存数据库TimesTen的专著

High Concurrency Oracle Database System Architecture and Design

高并发Oracle数据库系统的架构与设计

侯松◎著



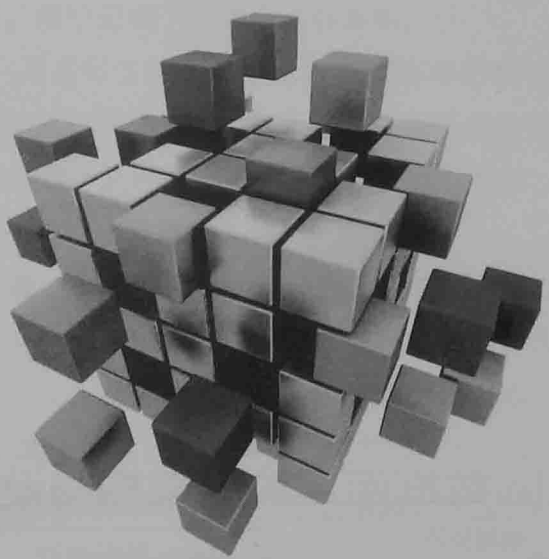
机械工业出版社
China Machine Press

数据库 技术丛书

High Concurrency Oracle Database System
Architecture and Design

高并发Oracle数据库系统的架构与设计

侯松◎著



机械工业出版社
China Machine Press

图书在版编目（CIP）数据

高并发 Oracle 数据库系统的架构与设计 / 侯松著 . —北京：机械工业出版社，2014.11
（数据库技术丛书）

ISBN 978-7-111-48227-7

I. 高… II. 侯… III. 关系数据库系统 IV. TP311.138

中国版本图书馆 CIP 数据核字（2014）第 236686 号

高并发 Oracle 数据库系统的架构与设计

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：高婧雅 责任校对：殷 虹
印 刷：三河市宏图印务有限公司 版 次：2014 年 11 月第 1 版第 1 次印刷
开 本：186mm × 240mm 1/16 印 张：24.25
书 号：ISBN 978-7-111-48227-7 定 价：69.00 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 投稿热线：（010）88379604
购书热线：（010）68326294 88379649 68995259 读者信箱：hzjsj@hzbook.com

Foreword 推荐序一

文以载道　书以自娱

侯松的新书付梓，嘱我为序，品读精华章节，览其前言，心有所感，遂言而记之。

关于写作之因由，于作者来说，一直是最为重要的缘起。

认真地写作一本好书，其中的坚持、勤勉、钻研对于作者来说都是极大的考验，所以古人说靡不有初，鲜克有终，侯松完成了这本书，恭喜他。

作者在前言中提及，其实写这本书是他内心积攒多年的情节所在，我也曾听其屡屡言及，虽有所期，但是未望其果。今日书成，方知作者坚持终未放弃，此诚可贵。

然而写作一本既可以自娱，又可以载道的作品就又难上加难，尤其是今日 Oracle 技术书籍几乎汗牛充栋，这一挑战绝对不容小觑。

纵观侯松大作，其结构设计堪称匠心独运，让读者眼前一亮，书中第一部分，于 Oracle 数据库的技术细节仅选取有关索引设计、表设计、优化器等核心部分，我相信这也是作者最有心得的部分，所谓授人以长，读者必能从他十数年的经验中体会 Oracle 的核心技术；而书的第二部分涉及了 Oracle 纵横扩展的技术体系，包括了 TT、GG、DG 等知识（国内书籍在 TT、GG 方面几乎空白），作者从架构入手，取 Oracle 延展的核心产品和解决方案，以自己在金融领域的实践为依托，娓娓演绎出来。

如此两两结合，内取 Oracle 核心技术，外取延展核心产品，理论与实践相结合，紧扣书旨之架构之意。

以此框架与立意而论，业内尚未有与此绝类之书，相信读者可以从中获得更为宏观的技术视野和实践经验，如此则不负作者孜孜之作，拳拳之义。

我从作者的认真和经验之中，收获良多，感谢作者。

盖国强

云和恩墨创始人，Oracle ACE 总监

推荐序二 *Forword*

还记得10年前，初次与侯松见面还是刚刚大学毕业进入公司时，他正端坐在台前认真地编写代码。彼时和他还不在一个部门工作，但是对他的技术和工作态度早有耳闻，后来有幸和他一起为公司开发ERP系统，他严谨而灵活的思维让我觉得他天生就应该属于IT。当时系统的后台数据库是Oracle，按照规定我们这样的菜鸟是不能直接操作数据库的，即使出现问题也要找外籍专家来解决，Oracle给了我们高大上的第一印象。

这家公司虽大，奈何对刚刚走出校园、渴望知识的我们在技术提升方面实在有限，几个人闲下来总在讨论如何找到出路，想来每个年轻人都会经过这个阶段。当时初生牛犊，没有经过太多的犹豫，选择了Oracle这条道路。时至今日看来，Oracle为我们带来的除了还算顺利的职场生涯，也使平时工作中的思维模式有所改变，Oracle架构体系之复杂，设计理念之先进，即使在分布式，开源技术大行其道的今天仍然在工作中给到我大量的启示，让我对于其他IT技术能够做到触类旁通。在这里也想告诉正在寻找方向的IT人，热爱你的选择，一条路走到黑，做到行业顶尖，一定会有回报的。

市面上Oracle书籍甚多，然抄文档、炒冷饭、二把刀翻译书层出不穷，能称得上原创和精品的寥寥无几，涉及金融保险业内一些技术解决方案的更是凤毛麟角，这本书在技术和行业经验方面一定能够给广大读者以帮助。

庄浩英

DeNA China运维总监

Preface 前　言

为什么要写这本书

写一本 Oracle 数据库方面的技术书籍，是我一个持续了四五年的想法。本着自我总结和快乐分享的初衷，不只一次地咨询过 eygle 大师关于写书的细节，eygle 大师也热情地予以指导。遗憾的是，总是因为这样那样的原因，这个想法迟迟不能落地。

2013 年的夏天，我有幸作为微博特使参与了甲骨文全球大会（Oracle Open World）上海站的活动，跟一位甲骨文的朋友闲谈中，不经意聊到了与 Oracle 数据库“共事”已经快十年了。朋友说我应该有不少心得了，鼓励我花一年的时间来做一个总结，可以写一本书分享给更多的朋友。“十年”是一个非常特别的东西，它彻底激发出我写书的热情。凌乱的思绪，不知该写些什么的时候，联想到再游十年未见西湖的感触：

云恸风摧山北暮，桥断平湖，西子颜如故。曲院风荷香暗渡，余晖昨日穿朱户。

月澹星稀闻浪住，对酒当歌，言莫愁时苦。意若随心晴若雨，谁知明日鸿归处？

对于技术人来说，杭州渐渐演变成技术之城，然而因为西湖，她应该是艺术的。正如以艺术之眼去欣赏 Oracle 数据库，不仅仅是纯技术活，更能发现其艺术之美。怀着一颗附庸风雅之心，我决定写一本具有一定实用价值的数据库架构设计和性能优化方面的书。

回顾十年技术之路，如大多数同行一样，一切都是从 OCP 认证开始的，没能赶上 8i OCP 的末班车，只好搭乘了 9i OCP 的头班车。如今认证不如以前受重视了，然而我一直认为 OCP 给我们提供了一个完整的基础知识体系，其价值不在于那一张纸而已。现在，DBA 工作内容逐步实现了流程化和模块化，一些初学者已经可以轻松地完成一些复杂架构的搭建，却时常会因为一些基础的概念性的东西而纠结不清，我会毫不犹豫地推荐他们去进行 OCP 教材的学习。只有建立自己的基础知识体系，才能主动地去思考问题，才能开始专职 DBA 之路。

在十年之前，有专职 DBA 的公司可以说少之又少，早期的 DBA 都是从开发转过来的，做的人多了，也就有专职 DBA 这个概念，进而很多不愿意写代码的人也纷纷投身其中。各

个公司也出于系统安全和精细分工的考虑，开始禁止 DBA 了解、熟悉业务，禁止 DBA 访问业务数据等，以至于现在很多 DBA 没有开发能力，也不懂得业务应用，仅仅是一个数据库技术的支持者，进而导致 DBA 被误读为夕阳职业。早期的 DBA 为什么能备受重视，不仅仅是因为物以稀为贵，更多的是因为有开发背景，了解业务流程，具备复合能力，这才是最可贵的。可以说不懂得 DBA 技能的开发不是好开发，不懂得开发的 DBA 不是好 DBA。

可喜的是，随着时间的发展，大家都开始意识到这个问题，于是数据库架构师的概念应运而生，他们是一群复合能力的拥有者，是开发人员和 DBA 之间的桥梁。然而，复合能力也是有较强的行业依赖性的，没有可以跨行业的万能复合，也没有能完全跨行业实现的万能数据库架构。我在本书的编写过程中，也是以我熟悉的金融行业为立足点，尽可能地兼顾全面阐述。

本书将给读者一个全新的视角，秉承大道至简的主导思想，只介绍高并发数据库架构设计中最值得关注的内容，不在于某种技能的分享，而致力于一种方法论的建立，希望能抛砖引玉，以个人的一些想法和见解，为读者拓展出更深入、更全面的思路。

本书特色

从技术层面上讲，本书首次介绍了国外已经成熟使用，而国内方兴未艾的 TimesTen（简称 TT）和 GoldenGate（简称 GG）。TimesTen 作为一款基于内存的关系型数据库，同时兼顾了内存处理的快速和传统数据库的稳定，更具备了与 Oracle 数据库的无缝衔接，对于日趋增长的数据库并发量和即时响应速度要求，其将大有用武之地；GoldenGate 一度被誉为数据复制、集成、迁移、容灾之神器，在国内也开始风靡，本书将揭示其核心特点及优势前景。另外，Oracle 数据库的技术细节仅选取有关索引设计、表设计、优化器等部分，只讲述最核心的技术，以激发读者的联想，进而衍化至繁。

从适合读者阅读和掌握知识的结构安排上讲，分为“内政篇”和“纵横篇”。“内政篇”，Oracle 技术方面的优化，很多资料只是介绍如何做，却鲜有介绍为何这样做，本篇将让读者知其所以然，也是作者自身经验的一些总结。“纵横篇”，TimesTen 对很多 Oracle 从业人员而言基本上是只闻其名，不知其味的。在与淘宝、支付宝同行交流中，大家都觉得好，但是因为不了解而不用。本篇将给大家展开一个活生生的 TimesTen 应用指导。

本书就是从 Oracle 内部扩展、纵向扩展（TimesTen）、横向扩展（GoldenGate、ADG）三个维度为读者展开讨论，深入分析，并提供相应的解决方案。

本书中一些实操的例子和章节，比较适合 DBA、程序员，可以成为工作手边书；架构设计和深入分析方面的章节，比较适合架构师、BA、IA，可以分享经验，拓展解决问题的思路；工具和产品的应用场景定位等，更适用于 CTO、CIO 对技术市场前景的把握。

读者对象

- 数据库架构师
- 数据库管理员
- 开发应用架构师
- 数据库开发人员
- 数据库相关的管理人员
- 其他对数据库技术感兴趣的人员

如何阅读本书

本书分为两大部分，共计 9 章内容。

第一部分为“内政篇”，着重讲解 Oracle 数据库内部的架构设计优化和扩展，包括以下 5 章内容：

第 1 章 从实际问题出发，讨论现下流行的架构设计思路，并提出大道至简的架构理念，以数据库森林体系为基础，展开高并发 Oracle 数据库的架构设计方法论。

第 2 章 详细介绍索引扫描方式，索引对排序过程的意义，索引设计的方法与技巧，并深入剖析索引树分裂生长原理和索引维护方法。

第 3 章 详细介绍表设计中字段类型选择、字段顺序、分区表使用、冗余手段、数据生命周期管理等问题，以及行链接、行迁移、碎片管理等表结构维护手段。

第 4 章 详细介绍了优化器的种类，配置管理方法，成本计算原理，统计信息的收集管理策略，执行计划管理，以及基于 Oracle 工具的数据库性能影响分析。

第 5 章 从实际案例出发，详细介绍锁相关问题（包括 Lock、Latch、Mutex 等）的高并发案例，以及 REDO 相关的 I/O、进程、等待事件的分析。

第二部分为“纵横篇”，着重讲解基于 Oracle 数据库的纵向和横向扩展的架构设计，包括以下 4 章内容：

第 6 章 详细、深入地介绍了 Oracle 的一款内存数据库 TimesTen 的前世今生、安装配置技巧、缓存应用、高可用架构设计，以及基础对象的设计与管理。并从实际运维出发，介绍 TimesTen 的典型监控方法，数据备份恢复与数据迁移方法。最后结合实际高并发场景，对 TimesTen 进行评估与对比测试。

第 7 章 展开介绍使用 GoldenGate 的安装配置和监控，并结合链路原理的实现，拓展数据集成平台和异构数据库群的设计思路。

第 8 章 主要围绕 Data Guard 功能，介绍了“T-1”交易数据库和 ADG 数据库两种新

型的应用方式，拓宽了 Data Guard 功能的使用范围，使得其不仅仅是典型容灾解决方案，更能为解决高并发问题发挥积极作用。

第 9 章 带给读者一个总结性的整体介绍，并简单介绍一些超出纯技术范围的软技巧。

其中，第 2 章、第 3 章、第 4 章和第 6 章为本书的重点章节，如果你没有充足的时间完成全书的阅读，可以选择性地进行重点章节的阅读。如果你是一位有着一定经验的资深人员，本书可能会是一本不错的案前书。然而，如果你是一名初学者，请在开始本书阅读之前，先进行一些数据库的基础理论知识的学习。

勘误和支持

由于作者的水平有限，编写时间仓促，书中难免会出现一些错误或者不准确的地方，恳请读者批评指正。如果您有更多的宝贵意见，欢迎您访问我的个人网站 http://www.housong.net 进行专题讨论，我会尽量在线上为您提供最满意的解答。同时，您也可以通过微博 @ 麻袋爸爸，或者邮箱 houalex@gmail.com 联系到我，期待能够得到你们的真挚反馈，在技术之路上互勉共进。

致谢

感谢 Oracle 官方文档，在写作期间提供给我最全面、最深入、最准确的参考材料，强大的官方文档支持也是其他数据库所无法企及的。

感谢大仙、子夜流星两位好友，在我写作出现迷惑的时候，予以帮助和指导。

感谢 Tom Kyte、Jonathan Lewis、Tanel Poder 等多位大师，以及 ITPUB 和 CSDN 社区的各位技术专家们的博客文章，每次阅读必有所获，本书也多处引用了他们的观点和思想。

感谢机械工业出版社华章公司的首席策划杨福川和编辑高婧雅，在近一年的时间中始终支持我的写作，你们的鼓励和帮助引导我顺利完成全部书稿。

特别致谢

最后，我要特别感谢我的太太 Liven 和女儿，我为写作这本书，牺牲了很多陪伴她们的时间，但也正因为有了她们的付出与支持，我才能坚持写下去。

同时，感谢我的父母、岳父岳母，特别是我的岳母，不遗余力地帮助我们照顾女儿，有了你们的帮助和支持，我才有时间和精力去完成写作工作。

谨以此书献给我最亲爱的家人，以及众多热爱数据库技术的朋友们！

侯松

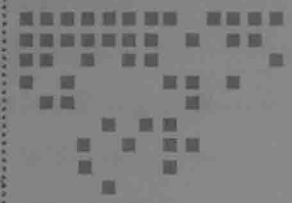
Contents 目　录

第一部分

内 政 篇

万物之始，大道至简，衍化至繁。

——老子，《道德经》



Chapter 1 第1章

大道至简

万物之始，大道至简，衍化至繁。世间一切事物在刚刚开始的时候，都是非常简单的，是为大道，随着事物不断地发展，其衍变出来各种复杂的局面。然而，追本溯源，仍然会发现复杂局面下真正在左右事物的还是那些最基本的东西——大道。如果能牢牢把握住这些被认作是“大道”的东西，再复杂的局面也自然会变得简单。

近些年来，在数据库领域里，可谓群雄逐鹿，志在中原。他们各自有各自的特色，相互间也甚是喜欢用自己的长处去比较对手的短处，以彰显其产品的优势，眼花缭乱的测试结果和市场评价，其局面之复杂就像将要开启一个新的时代。不论是作为研发者还是用户，这样的竞争无疑都是一件好事，但也要求我们需懂得把握其中的大道。

为什么会有如此多样的数据库在市场上竞争呢？当然是市场需求在推动的，而需求的本身则源自于各行业的业务系统应用。如果说未来会开启一个新数据库时代，那很好。然而，现在无疑还是 RDBMS 的时代，其中的佼佼者无疑还是 Oracle 数据库。

本书将秉承大道至简的主导思想，立足于 Oracle 的数据库相关技术，给读者展示一套高并发业务系统的数据库架构设计方法论。

1.1 初见高并发

高并发这个概念并不新鲜，可以说有数据库的地方都有可能面临高并发的问题。在数据库里，高并发问题主要集中在两个方面：读的高并发、写的高并发，两者看起来都不是很复杂，然而实际情况往往是读和写会交织在一起，并同时呈现出高并发的问题。

这个时候，相信很多读者都会提出一个观点：做读写分离嘛。是的，这是一个不错的主

意，但只是一个治标不治本的主意。业务系统的耦合度很高，是不可能实现业务层级的读写分离的。在架构设计的过程中，不能驻足于技术层面，还是需要渗透到业务层面去的。不论是业务驱动技术，还是技术驱动业务，两者都是不能分离开考虑的，比较好的做法应该是，在两者之间形成一种平衡，当然这是需要架构师的分析能力和沟通能力来加以驱动的。

1.1.1 从一次谈话说起

应用架构师："哥们！还记得上半年让你们帮忙做的一次数据库容量测试吗？"

数据库架构师："当然！我们可是费了很大的工夫才完成的。"

应用架构师："当时不是说我们在现有的硬件条件下，数据库的容量足够承受当时4倍业务压力吗？可是下半年我们的业务量才增长了0.5倍，怎么就不行了呢？"

数据库架构师："从系统资源利用率的报告来看，CPU、内存、I/O、网络上都没有太大的压力。但是，在业务高峰时段，CPU出现短暂冲高的现象。"

应用架构师："没错！在应用端可以看到出现用户阻塞，中间件上的连接持续增加，没有得到及时释放。"

数据库架构师："嗯，那是数据库未能及时响应造成的。CPU冲高，通过监控，我们发现很多并发的会话在争用一些资源，比如数据块、索引块、Latch和Mutex。表现出很多等待事件的冲高。"

应用架构师："我想我们遇到高并发争用的麻烦了。"

数据库架构师："是的。当初这套系统的设计，预期到现在这么大的业务压力了吗？"

应用架构师："谁也不会有这种先见之明吧？当初只是一种业务的尝试而已，谁能想到市场的反应会那么好！从数据库层面来看，有什么好办法吗？毕竟现在的问题出在数据库上。"

数据库架构师："办法是有的，针对性地一一解决掉。但是，这总归是治标不治本的办法。就像治水，现在压力我们堵住了，如果压力再大一些呢？我需要你的帮助。"

应用架构师："你是说业务架构的变更吗？小变化没有问题，但是要像互联网公司那样大改，肯定是不行的，我们接受不了这么大的变化。"

数据库架构师："是的，变革对我们来说，成本太大，小帆船掉头容易，航母掉头可能就要出事情了。"

应用架构师："我们最近在开发消息队列，能一定程度上缓解用户的高并发压力。但这是以牺牲用户体验作为代价的，终归不是太好的办法。"

数据库架构师："我想我们可以学习大禹治水，堵不上，就疏通。先将一部分业务压力分离出去，在子数据库和主数据库之间即时同步必要的数据，同时适当冗余数据到子数据库，减少数据库之间的交互。"

应用架构师："是个好办法，我们现在不就是这么在做的吗？问题不还是在那里吗？"

数据库架构师："我想是的。因为业务热点过于集中，在数据库上这部分业务数据耦合度又非常高，没有办法做到有效拆分。"

应用架构师："这不奇怪，我们的业务逻辑的复杂程度不是互联网应用可以比拟的，没办法像他们那样去充分地解耦。但是，我们是否可以学习他们使用廉价MySQL数据库代替Oracle数据库，这样业务分离的成本就低了很多。"

数据库架构师："你确定这样的成本会低吗？ Oracle数据库的开发人员真的会开发MySQL吗？我们的业务逻辑是很难在MySQL数据库上实现的，现在Oracle数据库帮助我们完成了大部分的数据耦合，如果使用MySQL，这些都将要在应用端完成，开发成本将会增加。另外，Oracle有强大的优化器机制，开发者写得不算太好的SQL，它也能包容，MySQL则不然了，SQL写得不好，马上给你颜色看，因为它的优化机制不能跟Oracle相提并论。与Oracle相比，MySQL只能作为一个数据容量来看待。"

应用架构师："你说的这些我也明白，但是我们的数据库是应该架构扩展了吧？"

数据库架构师："当然。虽然我们不能做到充分的数据解耦，但是一点一点地解还是可以的吧。我们现在的问题是解决高并发问题，而高并发集中的那部分业务就是我们的目标。"

应用架构师："那部分业务很难弄哦。读写比例为5：1，写操作占比还是蛮高的，而且都交织在一起的。最大的问题是要保证较快的响应速度，否则高并发压力就会积压。"

数据库架构师："内存数据库！将其作为Oracle数据库的前置缓存数据库来加快响应速度，如果保证了快速响应，高并发压力也就解决了。眼下Oracle和SAP都在内存数据库上大做文章呢。"

应用架构师："他们都是在搞OLAP的内存数据库吧，我们的可是OLTP。"

数据库架构师："Oracle的TimesTen就是为了解决OLTP的并发问题而研发的哦。关键是看我们的应用是否适应得了。"

应用架构师："嗯，是多样化的数据库架构了。"

数据库架构师："是的，就像一套数据库生态体系，没有必要特别待见谁，也没有必要特别排斥谁，关键是看应用的需要。不过，我们也需要反思一个问题，就是当初Oracle的设计阶段没有预见性，应该从Oracle内部设计先把握好，加强其处理并发的能力。是谓：'先修内政而后纵横。'"

1.1.2 问题就在那里

看过冗长的对话，我们来总结一下吧，问题到底出在哪里？