<<DB2数据库性能调整和优化>>

图书基本信息

书名：<<DB2数据库性能调整和优化>>

13位ISBN编号：9787302199533

10位ISBN编号：7302199531

出版时间：2009-5

出版时间：清华大学

作者：牛新庄

页数：499

前言

前　　言　　在介绍本书前，我先讲讲我的数据库学习之路。
我在1999年刚刚开始念硕士时就给自己确定了以后的发展方向，当时定了两个：网络和数据库技术。
因为2000年时，网络特别红火，拥有CCNP、CCIE认证的特别牛。
所以自己也考了CCNP证书，但是到后来我发现网络有很多是硬件层面的东西，对厂商的依赖特别强，而且面特别窄。
所以慢慢地就把这个方向放弃了，而我喜欢钻研，就选择了数据库技术。
在确认好数据库这个方向后，我深入地学习了数据库理论方面的知识，我记得中国人民大学王珊教授那本《数据库系统原理教程》一书我读了几十遍。
在对数据库理论学习的同时，我也开始对DB2和Oracle进行深入学习。
我是从1999年开始使用DB2 V5.2的，那时因为我导师做一个课题需要用到DB2数据库。
那时市场上关于DB2方面的技术书籍几乎没有，互联网还不像现在这么发达，自己只能依靠查看DB2随机文档来学习。
那时，我利用自己兼职帮别人做一些小软件和课题的费用去考OCP认证和DB2认证。
其实我认为考认证是一个很好的外界动力来促使自己学习，因为考试需要花费很大一笔费用，如果不想浪费钱就只能拼命地看书。
这是一个很大的促使自己看书的外界动力。
那时读研究生有的是时间，所以在2000年我就把OCP 8i的认证通过了，后来又陆续通过了DB2 V5.2的认证。
这些认证通过后能很大地增强自己的自信。
当时我同时帮导师做应用程序开发工作，那时用PB、Delphi等编程工具。
在开发中我有意识地增强自己对SQL的学习，这对我后来的性能调优非常有帮助。
因为我很多时候在客户现场看到同样一个操作，本来在数据库中利用函数或者其他高级SQL可以实现的，而开发人员却频繁地在数据库和应用程序之间进行切换，殊不知，在过程层(程序)和数据库层反复上下文切换交互会显著影响应用效率。
我认为自己应该是国内写SQL水平比较高的人，呵呵。
所以，首先要有一个清晰的方向和规划，然后有意识地去往这个方向努力。
做好一个时期的人生规划非常重要，它是你努力的方向，因为积极的学习比被动的学习效率要高太多。
机遇偏爱于有准备的人。
记得2001年初的时候我在网上看到一个帖子说要找一个人去安装DB2数据库，差旅报销，每天500元。
那时我就喜出望外，因为需要有DB2认证才能去，而我那时DB2系统管理、应用开发的认证都有，所以很快就通过了。
后来就去了客户现场，到了以后才发现不是安装，而是去给客户讲课，当时我就傻眼了，因为讲课需要的远比安装配置要难得多，而之前我没有讲过课。
没办法了，只能前一天夜里看教材备课到凌晨5点，睡两个小时，8点半去讲课，就这样4天讲课，我每天都是休息3小时左右。
还好自己毕竟使用过DB2，而且也过了DB2认证，还是有基础的。
这次讲课虽然不太成功，但是毕竟通过了，勉强可以打60分吧，没想到这次培训竟是我以后几年培训生涯的开始。
经过这次讲课后，我也看到了自己的差距，知道仅仅拥有认证是不行的，因为客户问的很多问题，书本上是没有的，说明自己还需要进一步努力，而且自己看书有些概念虽然不太懂也不会太深入研究，但是如果讲课你就自己必须把一些原理概念弄清楚，所以这需要对数据库有更深入的学习。
后来经过一些其他的渠道，IBM培训部知道我能讲DB2并且有相关证书，就找我讲DB2系列课程。
所以从2001年开始，我就经常作为IBM官方讲师开始讲DB2系列所有课程。
我感觉到讲课是个很好的学习过程，因为讲一些内容，你首先自己要搞清楚，这对提升自己有很大的

帮助，我把课堂学员问的实际产生的问题自己深入地研究。
我自己对培训有这样的认识，我是希望学员在这里听你讲3个小时要远远胜过自己看3个小时，而且如果把讲一堂课的内容比喻一杯水，那么老师至少应该储备一桶水，才能驾驭。
所以，我在讲课准备过程中，精心准备实验，深入和学员交流。
争取把一些概念能够用浅显易懂的例子来讲解，而要想做到这些，首先自己必须对这个概念有深刻的理解。
所以这在客观上促进了自己的学习。
　　随着培训的增多，也有一些客户找我去做一些实际的调优工作。
我记得第一次去客户现场调优是2001年去大连大通证券，当时主要解决锁等待问题。
客户环境用到了AIX和CICS环境。
当时虽然问题解决了，但自己心中还是感觉到比较虚，因为对AIX和CICS不了解，万一如果是这两个方面有问题，自己就没办法搞定了；同时认识到一个复杂系统的调整往往不是单方面的，需要具备全面的知识。
经过这次事情后，我就在网上买了一个140的IBM工作站小机，自己安装AIX，开始学习AIX。
这个期间我一边学习，一边把AIX的认证全部过掉。
我记得非常清楚，为了做HA的实验着实费了很大的功夫，因为无法搞到7133阵列，那时小型机不像今天这么普及。
后来自己又学习了CICS、WebSphere、MQ和存储，所以我认为有目的的学习，有压力和动力的学习效率是非常高的。
就这样，在我培训的过程中，我发现自己哪个方面薄弱并且这个方向有前途，我就开始学习，不过那个时候我的技术主要以IBM为主。
由于自己对培训比较用心以及客户的好评，开始找我做培训的国内培训机构开始变多，也感觉到这个期间自己的技术水平增长很快。
　　2002年11月，我参加首届“ IBM DeveloperWorksLive! China 2002 ”大会，并获得IBM首次在国内评选的“ 杰出软件技术专家 ”奖，在6名获奖者中名列第2。
获得这个认证后对我有很大的帮助，因为找我的人更多了，所以2002、2003年是我技术提升最快的两年，在这两年我又陆续学习了HP-UX、WebSphere和MQ并通过了认证。
其实我有时感觉到如果你把一门技术研究得非常深非常透，这时你再去学习另一门技术，就非常轻松，因为技术是触类旁通的。
我在学习完AIX再去学习HP-UX就感觉非常轻松。
同时，在学习Oracle和DB2后再去学习Informix感觉到很容易。
通过这种纵向的深入和横向的比较，就会思考它们的区别，就能够发现每一个产品的尺之所长、寸之所短，这样技术视野更加全局。
在学习过程中，不断地把实践和理论结合，不断地补充理论来充实自己，知其然知其所以然。
而且通过对一个产品的深入，往往能够发现这个产品的缺点和需要改进的地方。
就拿DB2来说，它的每次版本更新的新特性我基本上在新版本没有出来之前就猜得差不多了。
因为一是我是贴近真实用户的，了解他们的真实需求；二是自己一直在用，自己不断地总结思考；三是别的数据库有，而DB2没有，那么在下个版本就会增加。
所以相对来说，对新版本的新特性学习就非常轻松了。
　　就DB2而言，我是国内拥有DB2认证最多的人，我拥有DB2 V5.2、V7.1、V8.1和V9的全部认证。
我是国内第一个把DB2 V8认证全部通过的人，当然这也是巧合，因为2003年3月非典我被困在深圳，在网上看到DB2 V8认证从4月1日开始考，就在前一天报名注册，第二天把DB2 V8(新特性、高级DBA、应用开发、BI等所有认证)全部pass。
　　2004~2005年基本上是我最忙碌的两年。
那个时候找我讲课的培训机构以及性能调优的客户非常多，一年我基本上天天在天上飞(当然学校导师那边我是有办法搞定的)，培训机构找我讲课常常需要提前一个月预约时间。
那时除了过年在家几天，其他时间都是在做培训和诊断调优等，足迹踏遍国内主要城市。

那时我基本上是国内六大行开发中心和数据中心培训的指定老师，只要时间不冲突，还为北京银信科技、山东农信、广东农信、交行大集中IBP等项目做数据库技术顾问。
那时年轻很有精力，我记得有一次是2004年9月，我白天上午9点为上海移动IT部门做AIX动态逻辑分区(DLPAR)的培训；17点打车前往扬州，20点到达扬州供电局协助他们进行电力负荷控制系统项目上线，彻夜奋战到凌晨3点半；然后连夜打的赶往上海，凌晨6点到达酒店；休息两小时，8点出发，准时出现在上海移动培训现场。
那时我对报酬不太在意，我想的是拼命积累技术和客户资源，不断对技术学习、钻研、思考、提高，以及不断向上成长和孜孜不倦的探索，我就是这样在不断的学习、积累以及实践中成长起来的。
在很长一段时间内，我不断奔波于国内的各个城市，不计较出差、报酬，在我看来能够不断通过实践让自己成长是第一要义。
而且去的客户现场越多、处理的问题越多，就越发现自己的不足，然后就拼命地学习，不断地积累、总结和思考，这样就进入了一个良性循环。
至今我仍然怀念那种充实、紧张和激情的黄金年代。
2004年和2005年，我分别在上海、北京注册了公司。
一方面因为以独立咨询顾问的个人身份无法出具发票；另一方面，随着项目越做越大，尤其是很多银行的数据库架构和维护项目涉及合同金额也越来越大，需要签订合同，以公司的身份来签合同更加合适。
当然这些年并非所有都是一帆风顺，也犯过一些重大错误。
例如，我曾经在2002年5月1日把某机场的数据库调死，导致机场航班信息管理系统瘫痪；早期也曾经把证券系统因为调整而宕机。
这些都对客户造成了重大影响，同时也让我思考总结自己的不足。
也许这是成长必须要走的路。
所以，经过这两次事件之后，我之后的调优基本上没有犯过错误。
2006年8月我获得“2006年中国首届杰出数据库工程师”称号，也算是对我这么多年学习数据库的一个总结吧。
2007年开始，我专注于做一些大客户的运维，相应就减少了培训的次数。
2008年我被建设银行以217万年薪聘请为资深技术专家来维护Oracle和Informix数据库。
做技术而言，以一己之力能挣到年薪百万这常常是我感到自豪的地方，也是让我感觉到技术的魅力和自己这么多年对技术钻研的认可。
致学友　　其实我讲我的技术之路，主要给大家一些参考，尤其对在校学生，我希望我的学习之路大家能够模仿(毕竟大师级的路是很难模仿的)。
而且，这么多年，能取得这么一点小成绩，勤奋、努力和坚持一直是我看重的。
因为有了这些，不至于机遇惠顾你时，你怅然若失。
在现在的很多年轻人身上，我以为恰恰缺少的就是这样的忘我与痴迷。
在我熟悉的数据库技术领域，很多年轻人越来越早地将注意力集中在薪水和职位上，开始变得浮躁，而我想说的是，往往是那些将诸如高薪与职位忘怀的人反而能更快地提升。
不经一番寒彻骨，安得梅花扑鼻香。
这样的道理人人都懂，可是能够真正实施的人并不多。
所以，结合我的学习经验与感悟，有16个字送给进入这一领域的读者：去除浮躁，认真学习，不断积累，寻找机遇。
这些年我做数据库，相对于Oracle数据库而言，我深深感到DB2技术书籍的匮乏，所以我一直想写一套DB2方面的技术书籍。
坦白地说，写书是我挣钱性价比最低的一种方式，但是我一直感觉到自己有义务和责任去写一些东西来给大家分享，也算是对自己10年DB2学习经验的一个总结吧！
但是过去一直没有时间，直到今年奥运保障期间，有了一些时间，我才写了这套书。
在这套书中，我把应用开发和系统管理剥离开，分成两条线来讲解：《DB2应用开发实战指导》、《循序渐进DB2》、《深入解析DB2》，《DB2数据库性能调整与优化》。

其实关于DB2方面的技术还有很多我没有涉及，例如高可用性、数据库分区、HDAR、高级安全等，但是限于篇幅和时间，我无法做到全方位的解析。
关于本书　本书侧重在数据库的性能调优。
而性能调优是一个系统工程：全面监控分析操作系统、I/O性能、内存、应用及数据库才能快速找到问题根源；深刻理解DB2的锁及并发机制、索引原理、数据库参数、优化器原理、SQL语句调优等内部机理才能有针对性地快速提出解决问题的方法；快照、db2pd、db2expln及事件监控器等等则是必须要熟练掌握的工具。
本书正是覆盖了性能调优所需要的全部领域，并提供了大量的性能调优的实际案例。
本书系统性地总结了DB2数据库的性能调整的方法、流程、思路和保持系统良好性能的注意要点。
最后，与读者分享了我10年积累的DB2性能调优案例和经验总结。
本书所讲内容适用于DB2 V7、V8和V9.5的所有平台。
数据库学习之路　这里我对数据库学习做个总结：目前市场上虽然有Oracle、DB2、Informix、Sybase和SQL Server数据库，但是Informix数据库已经被IBM收购，而Sybase数据库在技术和市场上正走向没落。
那么剩下的其实就是Oracle、DB2和SQL Server数据库。
SQL Server数据库非常好，但是很遗憾的是只能在Windows平台使用，所以如果你深入研究SQL Server数据库我只能说你可以养家糊口，但挣大钱的概率小，因为用SQL Sever数据库的企业通常钱不多的，呵呵。
而国内做Oracle数据库的人太多了，如果你想在Oracle领域出人头地，难度极大。
反而DB2数据库做的人不太多，物以稀为贵。
DB2数据库广泛应用在银行、电信、制造、零售、保险等行业，所以我强烈建议你学习DB2数据库，钻研IBM技术一般相对来说挣大钱的概率会大些。
我们的时间精力是有限的，所以必须选择好方向然后努力为之。
古之成大事者，不唯有超世之才，亦唯有坚忍不拔之志也！
最后用这句话与大家共勉。
致谢　本书在出版的过程中得到了清华大学出版社王军编辑的大力支持！
这套DB2书籍从选题、审稿到出版无不得到他的热心帮助，在此致以深深的谢意！
感谢我的好兄弟骆洪青和袁春光，他们审核了书中的大部分章节。
同时也感谢中信银行的胡瑞娟、苏兰芳和我的师弟林春，他们审核了部分章节并从用户的角度给我提出了很多宝贵的建议！
最后，谨以此书献给我心爱的妻子，遇到她是我人生最大的成就！
新庄　于2009年新年
序二　2003年与新庄初次相识即给我留下了深刻印象，为人真诚、勤奋、踏实、富有热情，有着雄厚的数据库理论知识和高超的数据库技术，曾为大量的金融机构及大型企业作过数据库理论及操作培训。
他的授课深入浅出，将枯燥的理论知识讲得通俗易懂，赢得了所有培训人员和企业的高度肯定，为DB2数据库知识在中国的普及作出了巨大贡献。
在实务上，新庄更是帮助多家金融企业和机构解决了长期困扰的数据库性能问题。
和书店内满是Oracle数据库书籍不同的是，DB2数据库虽进入国内年数已久，但仅有几本IBM出版的介绍数据库基本操作的书籍，介绍性能调优、数据库内部机理机制等方面的书籍更是非常缺乏。
本人在金融行业软件开发以及系统优化的10年时间里，深刻感受到DB2相关书籍稀少的困苦。
偶与新庄谈起资料缺乏、高水平DB2书籍稀少的苦恼，均感慨DB2在金融行业使用如此广泛，大家在开发和运行优化时却仅靠口口相传，实在不符合DB2在国内金融行业的地位，也不利于这些核心系统的安全稳定运行。
因而突然看到新庄写了这样一本深入浅出、讲解极为透彻且在实务上也极具操作性的书，不由得喜出望外。
本书侧重于介绍DB2数据库的性能调优，而性能调优是一个系统工程：全面监测分析操作系统

、IO性能、网络传输、应用及数据库才能快速找到问题根源；深刻理解DB2的锁及并发机制、索引原理、数据库参数、优化器原理、SQL语句调优等内部机理才能有针对性地快速提出解决问题的方法；快照、db2pd、db2expln及事件监控器等则是必须要熟练掌握的工具。
这本书正是覆盖了性能调优所需要的全部领域，并提供了大量的性能调优的实际案例，为DB2开发、维护及系统优化人员都带来了福音，更为包括金融行业在内的众多核心系统的稳定运行及优化提供了指导利器。
本书的推广和热卖，我极具信心！
海通证券股份有限公司信息技术部总经理助理　　国内数据库顶尖高手　　曾任国内某证券行业TOP5软件公司技术总监　　曾主持开发期货交易所核心系统，并主持设计开发多家大型证券公司核心交易系统及风控等管理系统，在金融IT领域有着丰富的实践经验和专业背景知识，特别在大型数据库、大型系统架构设计等方面有着深厚的理论功底及实践经验。
王洪涛　　2009年1月　　序三(我与牛君初　　有些人，在你第一次见到时会蓦然想起四个字：相见恨晚。
我和牛新庄就属于这样的朋友。
这一次他完成了《DB2数据库性能调整和优化》一书，嘱我写点文字，我欣然允诺，虽然在DB2方面我是没有什么发言权的。
关于小牛　　我和小牛相识于2006年，在“中国首届杰出数据库工程师”评选活动中，我们入围了前十，同时获得了“杰出数据库工程师”的称号。
在最终评审的现场答辩之后，我们畅谈良久，探讨个人爱好、答辩主题、数据库技术，甚至职业发展等等。
在众多问题上的一致还并不是最重要的，最重要的是我们发现彼此的技术经历、个人性格等等竟然具有极大的相似性。
在这样的机缘巧合之下，我们成了朋友。
第一次见到小牛时，他穿白色衬衫，肩背黑色电脑包，一副行色匆匆、独闯天涯的形象，以至于多年之后，我记忆里的他还是依然如初。
谈到在面对技术的学习、钻研、思考、提高，以及不断向上的成长，我们都经历了孜孜不倦的探索过程。
小牛是在不断的学习、积累以及实践中成长起来的。
在很长一段时间内，他不断奔波于国内的各个城市，他不计较出差、也不计较报酬，在他看来能够不断通过实践让自己成长是第一要义。
同样，在这很长一段时间内，除了工作学习之外，其他一切都不在他的考虑范围之内。
就这样经历了在别人看来是痴迷、苦修一样的生活之后，终于成就了今日的一代大家。
在现在的很多年轻人身上，我以为恰恰缺少的就是这样的忘我与痴迷。
在我熟悉的Oracle技术领域，很多年轻人越来越早地将注意力集中在薪水和职位上，开始变得浮躁，而我想说的是，往往是那些将诸如高薪与职位忘怀的人反而能更快地抵达。
“不经一番寒彻骨，安得梅花扑鼻香”，这样的道理人人都懂，可是能够真正实施的人并不多。
所以，结合我们的经验与感悟，有16个字送给进入这一领域的读者：去除浮躁，认真学习，不断积累，寻找机遇。
关于本书　　在技术上，小牛几乎是无可挑剔的。
他涉猎极广，在数据库方面，对DB2、Oracle都具有精深的造诣，也正因为广博的知识与丰富的经验，他才能够敏锐地洞察不同数据库的尺之所短、寸之所长。
在和他关于技术的讨论中，经常让我受益匪浅。
市场上关于DB2的书籍本就不多，结合实践的经验总结就更是鲜见，小牛这本书无疑会给致力于学习与管理DB2数据库的技术人员们带来福音。
性能优化从来都不是单纯的事情，要想提高数据库的性能，必须综合考虑数据库、操作系统、存储甚至应用架构的设计，小牛将这些技术内容全面地涵盖在了本书之中，并且通过最后整整一章的内容来讲解实际案例的诊断过程，理论与实践相结合是最好的学习方法，这本书就是这样将精彩的内容

呈现给我们的。
关于勤奋　在技术道路上，我们一致认为能够与大家分享的最重要的两个字就是——勤奋。
也正是这两个字，才使我们成为惺惺相惜的好友。
我曾经在《循序渐进Oracle》一书的后记中写过一段关于勤奋的文字，转录在这里，因为这种勤奋的态度是我和小牛共同信奉的信条。
对于我个人，压力随时都在肩上。
而勤奋是我们最锋利的武器。
2005年，偶然读到王小慧的作品——《我的视觉日记》，感觉极为震撼。
那本书我认认真真地读过很多次，书的内容中，至今记忆犹新的是书尾列出的作者作品年表。
除了众多的摄影、电影拍摄与展览活动外，作者几乎保持了每年2、3本的作品出版速度，这需要多么大的勤奋付出以及坚持不懈啊！
有一位记者曾经这样写道：在她面前你会觉得惭愧，觉得自己至多是个中等水平的人，而且无可救药的懒惰。
这段话让我经常想起小牛，他比我年轻(所以我称他小牛)，但是其成就却常让我汗颜，他做学问至博士，做技术涉猎与范围又远较我广泛。
所以我经常以小牛为鉴，警示自己不可懈怠。
有一位可以为鉴的朋友，实为人生之幸运。
而关于勤奋，更让人敬佩的是另外一位大师——李敖，据说他曾经每月写作一本书，连续写了十年。
这些天才横溢的作家尚且如此至为勤奋，而我们，唯有更为努力。
所以，最后，虽然是在与文学完全不同的这条技术道路上，我仍然只有两个字作为最后的分享——勤奋。
独立咨询顾问，Oracle ACE 总监，《深入浅出Oracle》、《循序渐进Oracle》作者　盖国强(eygle)　2009年1月　序四　与牛新庄的接触源于10年前在大学里一起做项目，那时“恰同学少年”，虽无伟人们指点江山之意气风发，却也会因一个技术难题的攻克而一道去学校排档里大快朵颐。
牛新庄对技术追求之不懈在学校中就已经展现出那种“为伊消得人憔悴，衣带渐宽终不悔”的境界。
在21世纪初的狂热的互联网大潮中，牛新庄从纷繁复杂的IT技术中选择了数据库作为自己的主攻方向并且一直持续至今，是为一纲举而百目张，终于成为国内数据库顶尖级高手。
牛新庄涉猎极广，他从数据库出发，向下延伸至操作系统、存储，向上延伸至中间件，几乎所有企业应用涉及的平台他都有广泛而深入的研究，如AIX、HP-UX、IBM存储、EMC存储、Oracle、DB2、Informix、WebSphere、CICS、MQ等。
这些知识的融会贯通使得牛新庄在解决客户碰到的各种实际问题时，如庖丁解牛般游刃有余。
牛新庄在各种实践的基础上不断总结，能够从更高的视角反思DBA遇到的各种问题，并且上升到方法论，始有这套书的雏形。
DB2的学习资料在其信息文档和网络中有很多，但知识点分散，多不成体系，更是缺乏专家实践应用经验总结。
这套书凝聚了牛新庄大量的心血，是其10年DB2应用经验的总结。
这套书特点是注重实用，内容由浅及深，涵盖DB2的管理、运行维护、应用开发、内核及架构的剖析，以及性能调整和优化。
书中还有大量的提示点，虽只有寥寥数语，确是作者多年反复成功亦或失败的DB2实践体会，值得读者反复回味。
如果性能问题一直困扰着您，您也不要期望有一颗灵丹妙药，只需一试即能解决所有问题。
性能优化的问题，是对DBA综合能力的一个考验，需要DBA亲身实践去寻找答案。
但是在寻找答案的过程中您必须有一个正确、完整而且有组织的指引，否则这个答案您将永远找不到。
有人说看书是按图索骥，其实按图索骥又有何妨，至少您心里会有底，知道自己离目标的距离是远是

近。
读者如果能从本书这个“图”索到性能瓶颈这个“骥”，也就达到了作者授业、解惑之目标了。
北京银信长远软件技术有限公司 总经理 数据库高级专家 IBM官方资深培训讲师 骆洪青 2009年3月

<<DB2数据库性能调整和优化>>

内容概要

本书侧重于介绍DB2数据库的性能调优。
性能调优是一个系统工程：全面监控分析操作系统、I/O性能、内存、应用及数据库才能快速找到问题根源；深刻理解DB2的锁及并发机制、索引原理、数据库参数、优化器原理、SQL语句调优等内部机理才能有针对性地快速提出解决问题的方法；快照、db2pd、db2expln及事件监控器等则是必须熟练掌握的工具。
这本书正是覆盖了性能调优所需要的全部领域，并提供了大量的性能调优的实际案例。
本书系统性地总结了DB2数据库性能调整的方法、流程、思路和保持系统良好性能的注意要点。最难得的是作者分享了10年积累的DB2性能调优案例和经验总结。

<<DB2数据库性能调整和优化>>

书籍目录

第1章 性能调整概述 1.1 性能概述 1.2 性能评估 1.3 建立性能目标 1.4 什么时候需要做性能调整 1.5 性能调整准则 1.6 性能调整的方法和过程 1.6.1 性能调整的步骤 1.6.2 性能调整的限制 1.6.3 向客户了解情况 1.6.4 性能调整流程图 1.7 性能调整总结 第2章 存储I/O设计 2.1 存储基本概念 2.1.1 硬盘 2.1.2 磁盘阵列技术 2.1.3 存储的Cache 2.1.4 IOPS 2.1.5 网络存储技术 2.2 存储架构 2.2.1 存储I/O处理过程 2.2.2 应用系统I/O流动图 2.2.3 RAID IOPS 2.2.4 RAID 和RAID 的比较 2.3 存储相关性能调整案例 2.4 存储I/O设计总结 第3章 操作系统相关性能问题 3.1 HP-UX系统性能监控综述 3.1.1 监控资源对象和标准 3.1.2 监控工具 3.1.3 监控系统总体运行状态 3.1.4 性能状态的判定流程和监控命令 3.2 AIX性能监控综述 3.2.1 监控工具 3.2.2 监控系统总体运行状态 3.2.3 监控CPU性能 3.2.4 监控内存使用 3.2.5 监控存储系统状态 3.2.6 监控网络状态 3.3 操作系统性能优化 3.3.1 直接I/O和并发I/O 3.3.2 异步I/O和同步I/O 3.3.3 minpout和maxpout 3.3.4 文件系统和裸设备 3.3.5 负载均衡及条带化（Striping） 3.4 逻辑卷和lvmo优化 3.4.1 使用lvmo进行优化 3.4.2 卷组 pbuf 池 3.4.3 pbuf 设置不合理导致性能问题调整案例 3.4.4 使用 ioo 进行优化 3.5 总结 第4章 数据库物理设计和逻辑设计 4.1 数据库物理设计 4.1.1 表空间容器放置原则 4.1.2 数据库物理设计原则 4.2 数据库逻辑设计 4.2.1 缓冲池设计原则 4.2.2 表空间设计原则 4.3 使用Autoconfig设计数据库 4.4 其他高级设计技术 4.4.1 表分区及应用案例 4.4.2 数据库分区及应用案例 4.4.3 多维群集（MDC）及应用案例 4.4.4 物化查询表及应用案例 4.4.5 MDC、数据库分区、MQT和表分区配合使用 4.4.6 表压缩及应用案例 4.4.7 表压缩应用案例二 4.4.8 XML及应用案例 4.5 数据库设计总结 4.5.1 表空间与表设计方面的考虑 4.5.2 索引设计方面的考虑 4.5.3 缓冲池方面的考虑 4.5.4 总结 第5章 DB2性能监控 5.1 快照监视器案例 5.1.1 监控动态SQL语句 5.1.2 监控临时表空间使用 5.2 事件监视器及监控案例 5.3 利用表函数监控 5.4 性能管理视图及案例 5.4.1 监控缓冲池命中率 5.4.2 监控Package Cache大小 5.4.3 监控执行成本最高的SQL语句 5.4.4 监控运行最长的SQL语句 5.4.5 监控SQL准备和预编译时间最长的SQL语句 5.4.6 监控执行次数最多的SQL语句 5.4.7 监控排序次数最多的SQL语句 5.4.8 监控LOCK WAIT等待时间 5.4.9 监控LOCK CHAIN 5.4.10 监控锁内存使用 5.4.11 监控锁升级、死锁和锁超时 5.4.12 监控全表扫描的SQL 5.4.13 检查page cleaners是否足够 5.4.14 监控prefecher是否足够 5.4.15 监控数据库内存使用 5.4.16 监控日志使用情况 5.4.17 监控占用日志空间最旧的交易 5.4.18 用SQL监控健康指示器 5.4.19 监控存储路径 5.4.20 追踪监控历史 5.5 db2pd 5.5.1 常用db2pd监控选项和示例 5.5.2 使用db2pd监控死锁案例 5.5.3 db2pd使用问题总结 5.6 db2mtrk及监控案例 5.7 本章小结 第6章 数据库配置参数调整 6.1 数据库配置参数 6.2 监控和调优实例（DBM）配置参数 6.2.1 代理程序相关配置参数 6.2.2 SHEAPTHRES 6.2.3 FCM_NUM_BUFFERS 6.2.4 SHEAPTHRES_SHR 6.2.5 INTRA_PARALLEL 6.2.6 MON_HEAP_SZ 6.2.7 QUERY_HEAP_SZ 6.3 监控和调优DB配置参数 6.3.1 缓冲池大小 6.3.2 日志缓冲区大小（LOGBUFSZ） 6.3.3 应用程序堆大小（APPHEAPSZ） 6.3.4 SORTHEAP和SHEAPTHRES 6.3.5 锁相关配置参数 6.3.6 活动应用程序的最大数目（MAXAPPLS） 6.3.7 PKGCACHESZ 6.3.8 CATALOGCACHE_SZ 6.3.9 异步页清除程序的数目（NUM_IOCLEANERS） 6.3.10 异步I/O 服务器的数目（NUM_IOSERVERS） 6.3.11 组提交数目（MINCOMMIT） 6.3.12 AVG_APPLS 6.3.13 CHNGPGS_THRESH（DB） 6.3.14 MAXFILOP 6.3.15 LOGPRIMARY、LOGSECOND和LOGFILSZ 6.3.16 STMTHEAP 6.3.17 DFT_QUERYOPT 6.3.18 UTIL_HEAP_SZ （DB） 6.4 调整DB2概要注册变量 6.4.1 DB2_PARALLEL_IO 6.4.2 DB2_EVALUNCOMMITTED 6.4.3 DB2_SKIPDELETED 6.4.4 DB2_SKIPINSERTED 6.4.5 DB2_USE_PAGE_CONTAINER_TAG 6.4.6 DB2_SELECTIVITY 6.5 内存自动调优 6.5.1 内存自动调优示例 6.5.2 启用内存自动调优及相关参数 6.6 总结 第7章 锁和并发 7.1 锁等待及调整案例 7.1.1 锁等待问题解决流程和步骤 7.1.2 捕获引起锁等待的SQL 7.1.3 利用db2pd捕获锁超时 7.2 锁升级及调整案例 7.2.1 监控锁升级 7.2.2 锁升级调整

7.3 死锁及调整案例 7.3.1 利用事件监视器监控死锁 7.3.2 死锁案例 7.3.3 最小化死锁建议 7.4 隔离级别与锁 7.4.1 可重复读（RR—Repeatable Read） 7.4.2 读稳定性（RS—Read Stability） 7.4.3 游标稳定性（CS—Cursor Stability） 7.4.4 未提交读（UR—Uncommitted Read） 7.4.5 隔离级别加锁总结 7.4.6 隔离级别总结 7.5 最大化并发性 7.5.1 选择合适的隔离级别 7.5.2 尽量避免锁等待、锁升级和死锁 7.5.3 设置合理的注册变量 7.6 锁相关的性能问题总结 7.7 锁与应用程序开发 7.8 本章小结 第8章 索引设计与优化 8.1 索引概念 8.1.1 索引优点 8.1.2 索引类型 8.2 索引结构 8.3 理解索引访问机制 8.4 索引设计 8.4.1 创建索引 8.4.2 创建集群索引 8.4.3 创建双向索引 8.4.4 完全索引访问（index access only） 8.4.5 与创建索引相关的问题 8.4.6 创建索引示例 8.5 索引创建原则与示例 8.5.1 索引与谓词 8.5.2 根据查询所使用的列建立索引 8.5.3 根据条件语句中谓词的选择度创建索引 8.5.4 避免在建有索引的列上使用函数 8.5.5 在那些需要被排序的列上创建索引 8.5.6 合理使用INCLUDE关键词创建索引 8.5.7 指定索引的排序属性 8.6 影响索引性能的相关配置 8.6.1 设置影响索引性能的配置参数 8.6.2 为索引指定不同的表空间 8.6.3 确保索引的集群度 8.6.4 使表和索引统计信息保持最新 8.6.5 重组索引 8.7 索引维护 8.7.1 异步索引清除（AIC） 8.7.2 联机索引整理碎片 8.8 DB2 Design Advisor（db2advis） 8.9 索引调整总结 8.9.1 索引设计总结 8.9.2 索引性能总结 第9章 DB2优化器 9.1 DB2优化器介绍 9.2 SQL语句执行过程 9.3 优化器组件和工作原理 9.3.1 查询重写方法和示例：谓词移动、合并和转换 9.3.2 查询重写示例：视图合并 9.3.3 查询器重写示例：消除DISTINCT 9.3.4 查询器重写示例：隐含谓词 9.4 扫描方式 9.4.1 全表扫描 9.4.2 索引扫描 9.5 连接方法 9.5.1 嵌套循环连接 9.5.2 合并连接 9.5.3 哈希（hash）连接 9.5.4 选择最佳连接的策略 9.6 优化级别 9.7 如何影响优化器来提高性能 9.7.1 使DB2统计信息保持最新 9.7.2 构建适当的索引 9.7.3 配置合理的数据库配置参数 9.7.4 选择合适的优化级别 9.7.5 合理的存储I/O设计 9.7.6 良好的应用程序设计和编码 9.8 优化器总结 第10章 统计信息更新与碎片整理 10.1 统计信息更新 10.1.1 统计信息的重要性 10.1.2 统计信息更新示例 10.1.3 LIKE STATISTICS统计信息更新 10.1.4 列组统计信息更新 10.1.5 分布统计信息更新 10.1.6 统计信息更新策略 10.2 碎片整理 10.2.1 碎片产生机制和影响 10.2.2 确定何时重组表和索引 10.2.3 执行表、索引检查是否需要做REORG 10.3 重新绑定程序包 10.4 本章小结 第11章 SQL语句调优 11.1 通过监控找出最消耗资源的SQL语句 11.2 通过解释工具分析SQL语句执行计划 11.2.1 解释表 11.2.2 Visual Explain（可视化解释） 11.2.3 db2expln 11.2.4 db2exfmt 11.2.5 各种解释工具比较 11.2.6 如何从解释信息中获取有价值的建议 11.3 理解SQL语句如何工作 11.3.1 理解谓词类型 11.3.2 排序和分组 11.3.3 连接方法 11.3.4 扫描方式 11.4 SQL调优案例 11.4.1 用一条语句即可做到时避免使用多条语句 11.4.2 合理使用NOT IN和NOT EXISTS 11.4.3 合理使用子查询减少数据扫描和利用索引 11.4.4 调整表的连接顺序，减小中间结果集的数据量 11.4.5 在有偏差数据的情况下使用参数标记时，指定选择性 11.4.6 SQL使用UDF代替查询中复杂的部分 11.4.7 从多个SQL语句到一个 SQL表达式 11.4.8 使用SQL一次处理一个集合语义 11.4.9 在无副作用的情况下，请使用SQL 函数 11.4.10 小结 11.5 提高应用程序性能 11.5.1 良好的SQL编码规则 11.5.2 提高SQL编程性能 11.5.3 改进游标性能 11.5.4 根据业务逻辑选择最低粒度的隔离级别 11.5.5 通过REOPT绑定选项来提高性能 11.5.6 统计信息、碎片整理和重新绑定 11.5.7 避免不必要的排序 11.5.8 在C/S环境中利用SQL存储过程降低网络开销 11.5.9 高并发环境下使用连接池 11.5.10 使用Design Advisor（db2advis）建议索引 11.5.11 提高批量删除、插入和更新速度 第12章 DB2调优案例、问题总结和技巧 12.1 调优案例一：某移动公司存储设计不当和SQL引起的I/O瓶颈 12.2 调优案例二：某银行知识库系统锁等待、锁升级引起性能瓶颈 12.3 调优案例三：某汽车制造商ERP系统通过调整统计信息提高性能 12.4 调优案例四：某农信社批量代收电费批处理慢调优案例 12.5 调优学习案例：利用压力测试程序学习DB2调优 后记 参考文献

章节摘录

存储I/O设计　　对于任何程序的运行来说，最慢、最花费时间的操作实际上是从磁盘中检索数据。
这主要缘于磁盘 I/O 访问中存在的物理机械过程(磁头旋转和寻道)。
尽管磁盘存储技术在最近几年取得了极大的进步，但磁盘的旋转速度却没有太大的提高。
您必须清楚这样一个事实：在一定条件下，RAM 访问仅需要大概 540个 CPU时钟周期，而磁盘访问则需要花费大概20 000 000个CPU 时钟周期。
很明显，系统中访问数据最薄弱的环节就是磁盘 I/O 存储系统，从性能调整的角度来说，就是确保磁盘数据布局不会成为更严重的瓶颈。
糟糕的数据布局将会给 I/O 性能带来更大的影响。
在对系统进行任何优化活动之前，首先应该了解您的存储 I/O 系统的物理体系结构，因为如果您所设计的存储 I/O 系统非常糟糕，并且其中包含慢速磁盘，或者适配器的使用非常低效，那么其他的任何优化工作都无法提供帮助。
数据库的作用就是实现对数据的管理和查询。
任何一个数据库系统，必然存在对数据的大量读、写操作。
所以I/O问题也往往是导致数据库性能问题的重要原因。
要控制好数据库的整体I/O性能，在规划数据库架构时就需要做好存储I/O系统的设计和配置。
例如，将对I/O要求不同的文件放置在不同的存储设备上；规划表空间容器的分布、均衡I/O负担、使用并行I/O访问等。
在一个应用系统的逻辑部署和物理部署图中，存储和操作系统位于应用系统体系结构的最底层。
而数据库是部署在操作系统和存储层之上的，所以我们在做数据库的物理设计和逻辑设计之前，必须先做好存储I/O设计。
存储I/O设计中最大的一个原则就是将I/O访问的分布最大限度地平衡在所有可以利用的物理设备上。
本章主要内容包括：　　？存储基本概念　　？存储架构　　？存储相关性能调整案例　　？存储设计性能相关问题　　？存储I/O设计总结　　2.1　存储基本概念　　关于存储的概念太多，许多已经超出了本书的讨论范围。
本章主要讲解最常见的几个概念，它们是我们进行存储I/O设计所必须掌握的。
2.1.1　硬盘　　硬盘处于整个存储系统的最底层，核心的业务数据通常都存放在硬盘上。
我们从硬盘上读取一次I/O要花费的时间如下：　　硬盘上一次I/O时间=磁盘寻道时间+磁头旋转到特定扇区时间+传输时间+延迟　　图2-1所示的硬盘I/O传输图中标识了磁头在不同位置的寻道时间。
衡量一个磁盘的I/O能力有如下几个指标：　　图2-1　磁盘I/O传输图　　● 硬盘的转速(Rotational Speed)：也就是硬盘电机主轴的转速，转速是决定硬盘内部传输率的关键因素之一，它的快慢在很大程度上影响了硬盘的速度。
同时，转速的快慢也是区分硬盘档次的重要标志之一。
硬盘的主轴马达带动盘片高速旋转，产生浮力使磁头飘浮在盘片上方。
要将所要存取资料的扇区带到磁头下方，转速越快，等待时间也就越短。
因此，转速在很大程度上决定了硬盘的速度。
目前市场上常见的硬盘转速一般有5400rpm、7200rpm和15000rpm。
理论上，转速越快越好，因为较高的转速可缩短硬盘的平均寻道时间和实际读写时间。
但是转速越快发热量越大，不利于散热。
现在的主流硬盘转速一般为15000rpm以上。
● 平均寻道时间(Average Seek Time)：指硬盘在盘面上移动读写头至指定磁道寻找相应目标数据所用的时间，它描述硬盘读取数据的能力，单位为毫秒。
当单碟片容量增大时，磁头的寻道动作和移动距离减少，从而使平均寻道时间减少，加快硬盘速度。
● 平均延迟时间(Average Latency　Time)：指当磁头移动到数据所在的磁道后，然后等待所要的数据块继续转动到磁头下所用的时间。

● 平均访问时间(Average Access Time)：指磁头找到指定数据的平均时间，通常是平均寻道时间和平均延迟时间之和。
平均访问时间最能够代表硬盘找到某一数据所用的时间，越短的平均访问时间越好。
为什么要讲硬盘呢？
因为DB2数据库在工作时，一条SQL语句在经过优化器编译时，优化器会读取统计信息、数据库配置参数和相关硬件信息来为该条SQL语句生成最优的执行计划。
其中的硬件信息包括表空间的transrate和overhead。
这两个参数的计算就是由硬盘的相关属性来决定的。
它们计算公式如下： transrate=(1/传送速率)*1000/1024000*4096(假设用4KB页大小) overhead=平均寻道时间+(((1/磁盘转速)*60*1000)/2) 而平均寻道时间、磁盘旋转速度和传送速率是由硬盘本身决定的。
所以我们必须做合理的存储I/O设计以使优化器更好地工作。
2.1.2 磁盘阵列技术 RAID的全称是独立磁盘冗余阵列(Redundant Array of Independent Disks)。它通过将多个相对比较便宜的磁盘组合起来，并相互连接，同时都连到一个或多个计算机上，以组成一个磁盘组，从而使其性能和容量达到或超过一个价格更昂贵的大型磁盘。
20年来，RAID 推出了一系列级别，包括RAID 0、RAID 1、RAID 2、RAID 3、RAID4、RAID 5，以及各种组合如 RAID 1+0等，其中应用最广泛的是RAID5和RAID10。
RAID 1+0 RAID-0能提供更好的性能，RAID-1能提供最佳的数据保护。
如果把两者结合在一起，就能同时提供高性能和数据保护，但这也会同时提高磁盘阵列造价。
RAID-5 RAID-5不做全磁盘镜像，但它会对每一个写操作做奇偶校验计算并写入奇偶校验数据。
奇偶校验磁盘避免了像RAID-1那样完全重复写数据。
当一个磁盘失效，校验数据被用来重建数据，从而保证系统不会崩溃。
为避免磁盘瓶颈，奇偶校验和数据都会被分布到阵列中的各个磁盘。
尽管读的效率提高了，但是RAID-5需要为每个写操作做奇偶校验，因此它的写的效率很差。
2.1.3 存储的Cache 高端存储系统中除了要有高性能、高扩展性的结构外，Cache的设计将直接影响到存储系统的性能表现和可靠性。
Cache的主要作用是什么呢？
作为缓存，Cache的作用具体体现在读与写两个不同的方面：作为写，一般存储阵列只要求数据写到Cache就算完成了写操作，当写Cache的数据积累到一定程度，阵列才把数据刷到磁盘，这样可以实现批量的写入。
所以，阵列的写是非常快速的。
至于Cache数据的保护，一般都依赖于镜像与电池(或者是UPS)。
Cache在读数据方面的作用同样不可忽视，因为如果所要读取的数据能在Cache中命中的话，将大大减少磁盘寻道所需要的时间。
因为磁盘从开始寻道到找到数据，一般都在6ms以上，而这个时间，对于那些密集型I/O的应用可能不是太理想。
但是，如果能在Cache保存的数据中命中，一般响应时间则可以缩短在1ms以内。
存储的Cache大小对整个I/O性能的影响是非常大的。
2.1.4 IOPS IOPS，即I/O Per Second，也就是每秒进行读写(I/O)操作的次数，它是衡量存储性能的一个重要指标。
2.1.5 网络存储技术 FC SAN(Storage Area Network，存储区域网)是一个高速的子网，这个子网中的设备可以从您的主网卸载流量。
通常，SAN由RAID阵列连接光纤通道(Fibre Channel)组成，SAN同服务器和客户机的数据通信是通过SCSI命令而非TCP/IP来实现的，数据处理是“块级”(block level)。
SAN通过特定的互联方式连接若干台存储服务器而组成一个单独的数据网络，提供企业级的数据存储服务，如图2-2所示。

SAN是一种特殊的高速网络，连接网络服务器和诸如大磁盘阵列或备份磁带库的存储设备，SAN置于LAN之下，而不涉及LAN。
利用SAN，不仅可以提供大容量的存储数据，而且地域上可以分散，缓解了大量数据传输对局域网的影响。
SAN的结构允许任何服务器连接到任何存储阵列，不管数据存放在哪里，服务器都可直接存取所需的数据。
图2-2　SAN存储示意图　　NAS是Network Attached Storage(网络附加存储)的简称。
在NAS存储结构中，存储系统不再通过I/O总线附属于某个服务器或客户机，而是直接通过网络接口与网络直接相连，由用户通过网络访问。
它是连接到一个计算机网络的文件层的数据存储，可以为不同网络的客户端提供数据存储服务。
NAS的硬件与传统的专用文件服务器相似，它们的不同点在于软件端。
NAS中的操作系统和其他软件只提供数据存储、数据访问功能，以及对这些功能的管理。
与传统以服务器为中心的存储系统相比，数据不再通过服务器内存转发，而是直接在客户机和存储设备间传送，服务器仅起控制管理的作用。

版权说明