

M

计

算

机

科

学

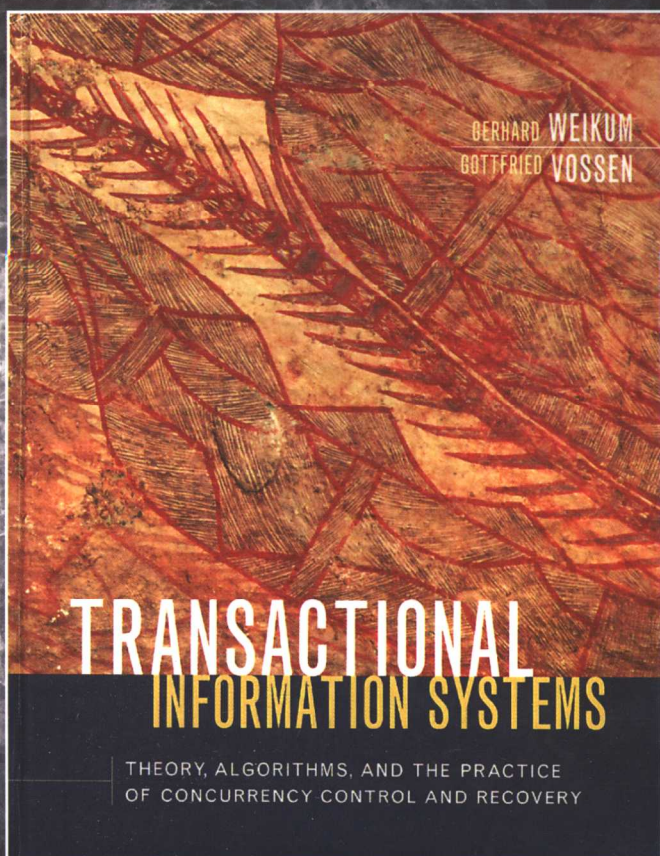
丛

书

事务信息系统

并发控制与恢复的理论、算法与实践

(德) Gerhard Weikum Gottfried Vossen 著 陈立军 邱海艳 赵加奎 等译



Transactional Information Systems

Theory, Algorithms, and the Practice of
Concurrency Control and Recovery



机械工业出版社
China Machine Press



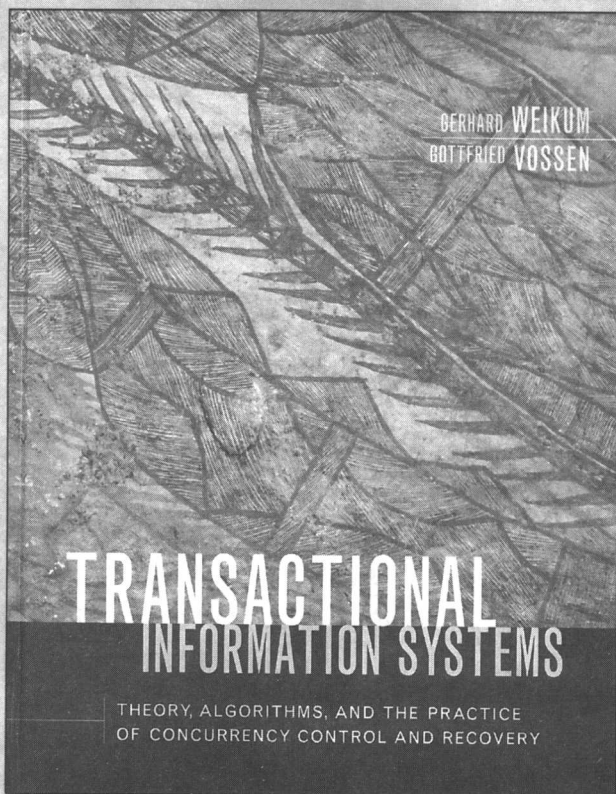
中信出版社
CITIC PUBLISHING HOUSE

计 算 机 科 学 丛 书

事务信息系统

并发控制与恢复的理论、算法与实践

(德) Gerhard Weikum Gottfried Vossen 著 陈立军 邱海艳 赵加奎 等译



Transactional Information Systems
Theory, Algorithms, and the Practice of
Concurrency Control and Recovery



机械工业出版社
China Machine Press



中信出版社
CITIC PUBLISHING HOUSE

本书从理论上介绍事务处理技术的方方面面，涵盖多版本并发控制，包括索引结构上的并发，事务恢复、崩溃恢复、介质恢复和应用恢复，分布式事务的并发与恢复等，同时也对在工作流以及电子商务应用中如何运用事务技术给出了参照标准。本书论述优雅紧凑，推理严谨洗练，是阐述对象模型上的事务处理方法的专著。

本书适合作为高等院校计算机等相关专业的教材或参考书，也可供专业技术人员参考。

Gerhard Weikum, Gottfried Vossen: Transactional Information Systems: Theory, Algorithms, and the Practice of Concurrency Control and Recovery (ISBN 1-55860-508-8).

Copyright © 2002 by Elsevier Science (USA).

Translation Copyright © 2006 by China Machine Press and CITIC PUBLISHING HOUSE.

All rights reserved.

本书中文简体字版由美国Elsevier Science公司授权机械工业出版社与中信出版社共同出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

版权所有，侵权必究。

本书法律顾问 北京市展达律师事务所

本书版权登记号：图字：01-2001-5272

图书在版编目（CIP）数据

事务信息系统：并发控制与恢复的理论、算法与实践/（德）威肯（Weikum, G.），（德）沃什（Vossen, G.）著；陈立军等译。—北京：机械工业出版社，2006.1

（计算机科学丛书）

书名原文：Transactional Information Systems: Theory, Algorithms, and the Practice of Concurrency Control and Recovery

ISBN 7-111-17531-X

I. 事… II. ①威… ②沃… ③陈… III. 数据库系统—理论研究 IV. TP311.131

中国版本图书馆CIP数据核字（2005）第116138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：朱起飞 冯春丽

北京瑞德印刷有限公司印刷·新华书店北京发行所发行

2006年1月第1版第1次印刷

787mm × 1092mm 1/16 · 31.75印张

印数：0 001-3 000册

定价：78.00元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换
本社购书热线：（010）68326294

出版者的话

文艺复兴以降，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了垄断性的优势；也正是这样的传统，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅擘划了研究的范畴，还揭橥了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短、从业人员较少的现状下，美国等发达国家在其计算机科学发展的几十年间积淀的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章图文信息有限公司较早意识到“出版要为教育服务”。自1998年开始，华章公司就将工作重点放在了遴选、移译国外优秀教材上。经过几年的不懈努力，我们与Prentice Hall, Addison-Wesley, McGraw-Hill, Morgan Kaufmann等世界著名出版公司建立了良好的合作关系，从它们现有的数百种教材中甄选出Tanenbaum, Stroustrup, Kernighan, Jim Gray等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及度藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力襄助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专程为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍，为进一步推广与发展打下了坚实的基础。

随着学科建设的初步完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都步入一个新的阶段。为此，华章公司将加大引进教材的力度，在“华章教育”的总规划之下出版三个系列的计算机教材：除“计算机科学丛书”之外，对影印版的教材，则单独开辟出“经典原版书库”；同时，引进全美通行的教学辅导书“Schaum's Outlines”系列组成“全美经典学习指导系列”。为了保证这三套丛书的权威性，同时也为了更好地为学校和老师服务，华章公司聘请了中国科学院、北京大学、清华大学、国防科技大学、复旦大学、上海交通大学、南京大学、浙江大学、中国科技大学、哈尔滨工业大学、西安交通大学、中国人民大学、北京航空航天大学、北京邮电大学、中山大学、解放军理工大学、郑州大学、湖北工学院、中国国家信息安全测评认证中心等国内重点大学和科研机构在计算机的各个领域的著名学者组成“专家指导委员会”，为我们提供选题意见和出版监督。

这三套丛书是响应教育部提出的使用外版教材的号召，为国内高校的计算机及相关专业

的教学度身订造的。其中许多教材均已为M. I. T., Stanford, U.C. Berkeley, C. M. U. 等世界名牌大学所采用。不仅涵盖了程序设计、数据结构、操作系统、计算机体系结构、数据库、编译原理、软件工程、图形学、通信与网络、离散数学等国内大学计算机专业普遍开设的核心课程,而且各具特色——有的出自语言设计者之手、有的历经三十年而不衰、有的已被全世界的几百所高校采用。在这些圆熟通博的名师大作的指引之下,读者必将在计算机科学的宫殿中由登堂而入室。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑,这些因素使我们的图书有了质量的保证,但我们的目标是尽善尽美,而反馈的意见正是我们达到这一终极目标的重要帮助。教材的出版只是我们的后续服务的起点。华章公司欢迎老师和读者对我们的工作提出建议或给予指正,我们的联系方法如下:

电子邮件: hzjsj@hzbook.com

联系电话: (010) 68995264

联系地址: 北京市西城区百万庄南街1号

邮政编码: 100037

专家指导委员会

(按姓氏笔画顺序)

尤晋元
石教英
张立昂
邵维忠
周克定
郑国梁
高传善
裘宗燕

王 珊
吕 建
李伟琴
陆丽娜
周傲英
施伯乐
梅 宏
戴 葵

冯博琴
孙玉芳
李师贤
陆鑫达
孟小峰
钟玉琢
程 旭

史忠植
吴世忠
李建中
陈向群
岳丽华
唐世渭
程时端

史美林
吴时霖
杨冬青
周伯生
范 明
袁崇义
谢希仁

译者序

历史总是处于不断的嬗变之中。许多时令的弄潮儿，有的被沙汰，湮没而不知所终，有的则经受住了考验，当时则荣，众口交传，其后便逐渐掩隐在历史的群山怀抱中了。正如村落中一口不起眼的井，司空见惯，好像生活中生就的一部分。可正是有了它，才使得上古之人摆脱了对河流和季节的依赖，能够定居下来，大大延拓了文明的范围。也许，只有小时曾经在油灯下苦读过的人才会激动于电灯的光芒。伟大发明所引起的那种震撼和欣悦甚至感动只存在于亲眼目睹它诞生的人们当中。

如果有人置疑信息技术是20世纪最伟大的发明，这并不奇怪，因为我们已经完全沉浸在基于信息系统所构建的世界里了。其实说它是20世纪最重要的发明一点也不为过，人类的一切文明活动，都将在信息系统的基础上展开。也许你午餐的内容仍旧会是一块比萨，但是现在这块比萨是按照你的口味定制的并在你方便的时间送到你的桌前。我们目睹这一变化，其中有些甚至参与推动了这一变化，这是属于我们的时代，我们为此激动不已。而对于我们的下一代来讲，在他们的理解当中，世界是生就如此的。

事务处理技术是有效管理和使用计算机信息资源的关键。透视我们日常生活的方方面面，我们在与银行、股市、航空公司、旅行社等打交道时，实际上都是基于一个个的事务处理方式。事务是一种“要么全部，要么全不”的语义范型，它的ACID特性可以保证数据在并发访问时的一致性以及发生故障时的可恢复性。如果在系统的某个基础层次上实现了事务的概念，则上层应用逻辑的处理就会变得统一而且简单。因此事务处理技术不但在传统上用于操作系统、数据库系统中，而且它使得一些事务型资源管理器如队列管理器、工作流系统等也變得日益流行。事务处理已经变成现代信息系统的基础支撑部件，随着电子商务应用以及基于Web的电子服务的飞速发展，它变得益发重要。

本书共由五部分构成，第一部分概述事务处理的背景与动机，并从理论上讲解了所基于的计算模型：页模型和对象模型；第二部分和第三部分是全书的核心，分别讲述了事务处理的两个基本部件：并发和恢复，涵盖了（多版本）并发控制，包括索引结构上的并发、事务恢复、崩溃恢复、介质恢复和应用恢复等一系列相关内容。第四部分讲述有关分布式事务的并发控制和事务恢复，以及在工作流和电子商务应用中运用事务技术的考虑；第五部分则是应用与未来展望。需要提及的是，这是第一本阐述对象模型上事务处理的教材。

本书是典型的德国人的著作，遣词严谨工整，行文雅驯而不拗口，叙述广博而不枝蔓，可谓一本哲学风格的工程技术书。读者可以将其与Jim Gray的《事务处理：概念与技术》一书相对照，相信两者会相得益彰。后者偏重于系统实现，是作者在工程实践领域浸淫多年的心得之作，可谓厚积而薄发，挥挥洒洒，优裕自如，概成大观。本书则是在作者对近30年来事务领域内的研究成果进行归纳综合的基础上精心结撰而成的，它强调理论阐述的高度形式化，以严谨洗练的推理方式给出了优雅紧凑的理论结果，蕴含着一种数学之美。当然这种推理并不是繁文缛节的，也是有着其深刻的应用背景因素的。相信读者经过潜心阅读，在掌握事务处理相关知识的同时，还能从中学到一套计算机科学领域的形式化推理方法。

参加本书翻译的还有张远、赵静和陈瑞怡，董炜作为本书正式出版前的第一位读者，对其中一些翻译得生硬拗口和难于索解之处提出了很好的建议，对于他们的辛勤工作，在此一并表示感谢。

正如图灵奖获得者Jim Gray所言，相信本书在未来很长一段时间内会被奉为事务处理领域内的圭臬。限于译者水平，原书精髓难于传神到位，虽经译者反复校核，错讹之处在所难免，欢迎广大读者批评指正，以便本书再版时勘误补遗。

陈立军

2005年8月

序

本书标志着事务处理技术的一个重要进步。它把近30年来领域内的研究成果进行综合组织，形成了严格一致的描述。它统一了页模型和对象模型的并发控制与恢复。与众多的参考文献一样，这种统一包含了除Weikum和Vossen以外的很多其他研究人员的工作；但是，本书把所有这些研究成果组织起来形成了一个完整一致的文集，并且以逐步深入的方式对一些想法和内容进行了讲述。

事务处理的优秀著作大都或者关注事务的实际应用，或者用严格的方法描述定理并进行证明。大多数限于“平面”事务，因为多级事务的理论是如此不成熟。本书是对事务相关领域的理论和实践进行深入描述以及讲述多级（对象模型）事务处理的最新进展的第一本书。

在读书的过程中，这一研究领域的发展以及采用恰当的术语和观察角度简单描述复杂的想法都给我留下了深刻的印象。你可以停留在表面来阅读本书：只是跟随书的文本进行阅读，包括例子、定义和定理。你也可以按照你喜欢的任意深度对本书中给出结果的详细描述进行阅读——包括证明和程序。在纵览本书的过程中，我采用了两种方法来阅读：把本书当成一本参考书，深入到某些章节中，看用多快的速度能找到问题的答案；我也按照顺序的方法对本书进行了阅读。从以上两种情况的结果来看，本书都是信息量极为丰富且非常容易理解的。

本书在未来多年内极有可能成为我们研究领域内的标准参考书。

Jim Gray, 微软公司

前 言

Teamwork is essential. It allows you to blame someone else.

—Anonymous

本书的使命

近30年来，事务处理成为现代信息技术发展过程中的一个里程碑：因为它成了银行业、股票交易、航空公司、旅行社等各个行业业务中不可缺少的一部分。在新的世纪里，随着电子商务应用、B2B工作流以及基于Web的电子服务的飞速发展，事务信息系统变得更为重要。幸运的是，事务这个概念的成功之处不仅在于精巧的系统实现，还在于研究团体为其提供的科学严格的理论基础，为事务信息处理的发展起到了很好的推动作用。理论基础中最显著的研究成果是由Jim Gray做出的，由于在事务概念上突出的开创性工作，他获得了1998年的图灵奖。正是基于对事务概念基本的、系统的理解，使我们可以对事务信息系统进行扩展和推广，使其能够适应以网络为中心的、功能更加丰富的应用。

由于以上的原因，本书着重讲述长期有效并且有价值的关于事务的理论基础，而不涉及具体的系统产品，因为产品很可能很快就过时了。然而，本书在实际的系统体系结构中讨论了现有的理论、算法以及实现技巧。从这一点上来讲，本书是对那些面向系统的著作的补充。在所有基于系统的著作中，最值得注意的是Jim Gray和Andreas Reuter所著的“事务处理圣经”《事务处理：概念与技术》（中译本已由机械工业出版社出版）以及最近由Phil Bernstein和Eric Newcomer所著的教科书。本书的功能模型与Phil Bernstein、Vassos Hadzilacos和Nat Goodman已绝版的经典著作《数据库系统的并发控制与恢复》（Concurrency Control and Recovery in Database Systems）是一致的。然而，从该书写成到现在，这个领域又有了长足的发展，事务的概念被拓宽到数据库系统以外的其他相关领域。我们这本书反映了过去10年间这个领域的发展和现代IT体系结构的发展方向。

本书的组织

事务信息系统的两个主要组成部分是并发控制（在很多客户端同时访问共享数据的时候确保数据库的正确性）与恢复（在发生系统故障的时候保护数据）。本书的两个重要组成部分，第二部分和第三部分，一共15章，着重讲述这两个组成部分。对于分布式的多级事务服务器的联合，我们将看到，每个事务服务器的并发控制与恢复部件是实现可行解决方案的主要资源；但是，“分布式事务之间的协调”是另外一个非常重要的问题，我们将在第四部分对其进行描述。对上述三个技术性的“核心”部分的介绍分别包含在第一部分和第五部分中，其中第一部分包含了背景与动机，并且给出了事务技术的“框图”；第五部分对本书没有讲述到的一些主题（由于时间和篇幅所限）给予展望并对未来的发展方向进行推测。贯穿所有五个部分，每章由摘要开始，给出该章的目标和概述，由小结、习题和文献注释结束（注意：第20章没有小结和习题）。

教学方案

本书包含很多高级素材，包括形式化模型的密集应用。因此，这里假设读者有坚实的计算机科学理论基础和背景，但是，不熟悉数据库系统也没关系。需要应用到的数据库系统相关领域的知识在本书中都给予了讲述。实际上，我们这样做的要点在于事务技术在很多其他领域也非常重要；例如，操作系统、工作流管理、电子商务以及分布式对象。因此，应该独立于数据库课程进行讲述。

编写本书最初的目的是作为高年级本科生或研究生相关课程的教材，同时，我们也鼓励需要深入理

解事务信息系统的业界的研究人员、系统构建人员以及开发人员学习本书。毕竟，工程师不应该恐惧（或者有一点点恐惧）数学。

本书已经在课堂教学中进行了测试，测试是在德国的几所大学进行的，包括在Saarbrücken的Saarland大学，Münster大学，还在Constance大学进行了部分测试；测试都是作为本科生的高级课程进行的。在Saarbrücken，课程组织为15个教学周，每周4小时的宣讲再加上额外的课后作业。下面给出了一个包含本书章节内容的教学时间表。由于很多大学对于这样的高级课程每周只允许有2个小时的讲授时间；因此，可以把本书的素材分成必修的核心主题和可选的“高端”主题，像表1给出的那样（每章的第一节和最后一节都是必修章节，因此这个表中省略了这两节）。

表1 15周课程的建议教学时间表

周	必修章节	可选章节
1	第1章: 概述 第2章: 计算模型 1.2~1.5, 2.2~2.5	
2	第3章: 并发控制 3.2~3.5, 3.7~3.8	3.6, 3.9~3.10
3	第4章: 并发控制算法 4.2, 4.3.1~4.3.4, 4.4~4.5	4.3.5~4.3.8
4	第5章: 多版本并发控制 5.2, 5.3.1~5.3.2, 5.5	5.3.3, 5.4
5	第6章: 对象上的并发控制: 正确性概念 6.2~6.5	6.6
6	第7章: 对象上的并发控制算法 第8章: 关系数据库的并发控制 7.2~7.5	7.6, 8.2~8.4
7	第9章: 搜索结构上的并发控制 第10章: 实现和实用性问题 9.2~9.3, 9.4.1, 10.2~10.3	9.4.2, 9.4.3, 9.5, 10.4~10.7
8	第11章: 事务恢复 11.2~11.4, 11.5.1, 11.6~11.7	11.5.2~11.5.3
9	第12章: 崩溃恢复: 正确性概念 第13章: 页模型崩溃恢复算法 12.2~12.4, 13.2, 13.3.1~13.3.3, 13.4	12.5, 13.3.4
10	第14章: 对象模型的故障恢复 14.2~14.4	14.5
11	第15章: 恢复的特别问题 第16章: 介质恢复 16.2.1~16.2.2, 16.3.1	15.2~15.6, 16.2.3, 16.3.2, 16.4
12	第17章: 应用恢复 17.2~17.4	17.5
13	第18章: 分布式并发控制 18.2~18.3, 18.4.1~18.4.2, 18.5	18.4.3, 18.6~18.8
14	第19章: 分布式事务恢复 19.2~19.3	19.4
15	第20章: 下一步是什么 20.2~20.5	

也可以根据本书的素材配置一个10周的授课计划。在这种时间约束下，显然需要省去一些高级的主题。我们在表2中给出了一个10周授课（每周2小时或4小时授课时间）的建议。

表2 10周课程的建议教学时间表

周	必修章节
1	第1章: 概述 第2章: 计算模型 1.2~1.4, 2.2~2.5
2	第3章: 并发控制 3.2~3.5, 3.7~3.8
3	第4章: 并发控制算法 4.2, 4.3.1~4.3.4, 4.4~4.5
4	第5章: 多版本并发控制 5.2, 5.3.1~5.3.2, 5.5
5	第6章: 对象上的并发控制: 正确性概念 第7章: 对象上的并发控制算法 6.2~6.5, 7.2~7.4
6	第10章: 实现和实用性问题 第11章: 事务恢复 10.2~10.3, 11.2~11.4, 11.5.1, 11.6~11.7
7	第12章: 崩溃恢复: 正确性概念 第13章: 页模型崩溃恢复算法 12.2~12.4, 13.2, 13.3.1~13.3.3, 13.4
8	第14章: 对象模型的故障恢复 第16章: 介质恢复 14.2~14.4, 16.2.1~16.2.2, 16.3.1
9	第17章: 应用恢复 第18章: 分布式并发控制 17.2~17.3, 18.2.1~18.2.2, 18.4.1~18.4.2, 18.5
10	第19章: 分布式事务恢复 19.2~19.3

额外的教学素材,最主要的是授课幻灯片和部分习题的解答,可以在www.mkp.com/tis/得到。一旦发现书中有错误,我们就将其加入本书的勘误表中。我们也非常欢迎对本书的评论、建议和批评,这些信息可以通过电子邮件weikum@cs.uni-sb.de或vossen@uni-muenster.de提交给我们。

致谢

很多同行给我们提供了非常有价值的东西:鼓励、建设性的批评、校对、课堂测试和简单的错误校正。我们非常感谢Jim Gray、Elliot Moss、Dennis Shasha、Betty和Pat O'Neil、K.Vidvasankar、Alan Fekete、Dave Lomet以及Marc Scholl。关于本书习题的重要工作是由Ralf Schenkel完成的;Carolyn Letz编辑和查证了很多参考文献条目。本书也潜在地受益于很多技术性的讨论以及与Catriel Beerli、Yuri Breitbart、Theo Härder、Dave Lomet,尤其是Hans-Jörg Schek的合作。当然,本书所有的偏见和可能的错误都是我们自己的过错。

我们的编辑Diane Cerra和她的同事Belinda Breyer也非常出色,因为她们在保持我们放松而有创造力的同时偶尔给我们一些健康的压力,而且在两者之间做到了很好的平衡,她们对我们的需要一直都给予支持。我们希望所有写书的人都能够有这样一个出色的编辑队伍。我们也非常幸运有机会和Edward Wade所领导的出色的出版队伍合作,包括我们的版本编辑Judith Brown,校对者Jennifer McClain,设计师Rebecca Evans和Frances Baca,索引编辑Steve Rath。最后,但相当重要的是,我们还要感谢我们的家人,当我们专注于工作而忽视了他们的时候,他们还陪伴在我们的身边。

目 录

出版者的话	
专家指导委员会	
译者序	
序	
前言	

第一部分 背景与动机

第1章 概述	1
1.1 目标和概述	1
1.2 应用举例	2
1.2.1 联机事务处理: 借/贷的例子	2
1.2.2 电子商务的例子	5
1.2.3 workflow管理: 旅行计划的例子	6
1.3 系统范型	8
1.3.1 三层体系结构和两层体系结构	8
1.3.2 服务器的联合	11
1.4 事务概念的优点	12
1.4.1 事务特性与事务编程接口	12
1.4.2 事务服务器的功能需求	14
1.5 数据库服务器的概念与体系结构	14
1.5.1 数据库系统的分层体系结构	14
1.5.2 数据是如何存储的	16
1.5.3 数据是如何被访问的	17
1.5.4 查询与更新是如何进行的	19
1.6 小结	21
习题	21
文献注释	21
第2章 计算模型	23
2.1 目标和概述	23
2.2 计算模型的组成部分	23
2.3 页模型	24
2.4 对象模型	27
2.5 本书的“路线图”	30
2.6 小结	31
习题	31
文献注释	32

第二部分 并发控制

第3章 并发控制: 页模型正确性的概念	33
3.1 目标和概述	33

3.2 经典的并发问题	33
3.3 历史和调度的语法	35
3.4 历史和调度的正确性	39
3.5 调度的Herbrand语义	40
3.6 终态可串行性	42
3.7 视图可串行性	45
3.7.1 视图等价和结果正确性准则	46
3.7.2 检测视图可串行性的复杂性	47
3.8 冲突可串行性	51
3.8.1 冲突关系	51
3.8.2 CSR类	52
3.8.3 冲突和交换性	54
3.8.4 冲突可串行性的约束	56
3.9 提交可串行性	57
3.10 一个可选的正确性准则: 交叉存取 说明	60
3.11 小结	65
习题	66
文献注释	67
第4章 并发控制算法	69
4.1 目标和概述	69
4.2 通用调度器的设计	69
4.3 锁调度器	72
4.3.1 简介	72
4.3.2 两阶段封锁协议	74
4.3.3 死锁处理	77
4.3.4 2PL的变体	79
4.3.5 有序的共享锁	80
4.3.6 利它锁	83
4.3.7 非两阶段封锁协议	86
4.3.8 封锁的几何学意义	89
4.4 非封锁调度器	91
4.4.1 时间戳排序	91
4.4.2 串行化图的检测	92
4.4.3 乐观协议	94
4.5 混合协议	96
4.6 小结	98
习题	99

文献注释	100	8.3.4 冲突可串行性	161
第5章 多版本并发控制	101	8.3.5 扩展的冲突可串行性	162
5.1 目标和概述	101	8.3.6 在函数依赖面前的可串行性	163
5.2 多版本调度	101	8.3.7 小结	165
5.3 多版本可串行性	103	8.4 应用事务程序知识	165
5.3.1 多版本视图可串行性	103	8.4.1 范例	166
5.3.2 MVSR成员资格检测	105	8.4.2 事务分割	167
5.3.3 多版本冲突可串行性	107	8.4.3 切割的适用性	169
5.4 限制版本的数目	109	8.5 小结	171
5.5 多版本并发控制协议	110	习题	171
5.5.1 MVTO协议	110	文献注释	173
5.5.2 MV2PL协议	111	第9章 搜索结构上的并发控制	174
5.5.3 MVSGT协议	114	9.1 目标和概述	174
5.5.4 只读事务的多版本协议	115	9.2 B+树搜索结构的实现	175
5.6 小结	116	9.3 访问层的键范围封锁	178
习题	116	9.4 页层的技术	183
文献注释	117	9.4.1 锁耦合	184
第6章 对象上的并发控制: 正确性概念	119	9.4.2 链接技术	189
6.1 目标和概述	119	9.4.3 放弃技术	190
6.2 历史和调度	119	9.5 进一步的优化	191
6.3 平面对象事务的冲突可串行性	122	9.5.1 无死锁的页门锁	191
6.4 树可归约性	125	9.5.2 增强的键范围并发	191
6.5 树可归约的充分条件	128	9.5.3 降低封锁开销	192
6.6 采用基于状态的可交换性	132	9.5.4 利用暂态版本化	193
6.7 小结	135	9.6 小结	193
习题	136	习题	194
文献注释	137	文献注释	195
第7章 对象上的并发控制算法	139	第10章 实现和实用性问题	196
7.1 目标和概述	139	10.1 目标和概述	196
7.2 平面对象事务封锁	139	10.2 锁管理器的数据结构	196
7.3 分层锁	140	10.3 多粒度封锁和动态提升	197
7.4 通用事务森林上的封锁	144	10.4 暂态版本化	199
7.5 混合算法	146	10.5 事务内部并行的嵌套事务	201
7.6 为返回值的可交换性加锁和契约锁	147	10.6 调整选项	201
7.7 小结	150	10.6.1 手动封锁	202
习题	151	10.6.2 SQL的隔离级别	202
文献注释	151	10.6.3 短事务	204
第8章 关系数据库的并发控制	153	10.6.4 多道程序级别的限制	206
8.1 目标和概述	153	10.7 过载控制	207
8.2 面向谓词的并发控制	154	10.7.1 反馈驱动方法	208
8.3 关系的更新事务	157	10.7.2 等待深度限制	210
8.3.1 语法和语义	158	10.8 小结	210
8.3.2 可交换性和简化规则	159	习题	211
8.3.3 历史和最终状态的可串行性	160	文献注释	211

第三部分 恢 复

第11章 事务恢复	213	13.3.3 增强算法: 日志截断、检查点、重做优化	269
11.1 目标和概述	213	13.3.4 完整的算法: 处理事务中止和撤销完成	281
11.2 带有显式Undo操作的扩展调度	214	13.4 重做历史范型	288
11.2.1 概念的直觉和概述	214	13.4.1 正常操作期间的操作	288
11.2.2 形式化模型	214	13.4.2 简单的三趟算法和两趟算法	288
11.3 页模型的正确性准则	216	13.4.3 增强的算法: 日志截断、检查点和重做优化	294
11.3.1 扩展冲突可串行性	216	13.4.4 完整的算法: 处理事务回滚和撤销完成	294
11.3.2 可归约性与前缀可归约性	217	13.5 小结	299
11.4 充分的句法条件	219	习题	306
11.4.1 可恢复性	220	文献注释	308
11.4.2 避免级联中止	220	第14章 对象模型的故障恢复	309
11.4.3 严格性	221	14.1 目标和概述	309
11.4.4 严厉性	221	14.2 重做历史算法的概念综述	309
11.4.5 日志可恢复性	224	14.3 一个简单的两层系统的重做历史算法	311
11.5 带有事务中止的页模型调度协议	227	14.3.1 正常操作期间的操作	312
11.5.1 为实现严格性和严厉性扩展两阶段封锁协议	227	14.3.2 重启期间的操作	313
11.5.2 为日志可恢复性扩展串行图检测	227	14.4 一个增强的两层系统的重做历史算法	316
11.5.3 为日志可恢复性扩展其他协议	229	14.5 一个完整的通用对象模型执行的重做历史算法	322
11.6 对象模型的正确性准则	229	14.6 小结	324
11.6.1 平面对象调度中的中止	229	习题	325
11.6.2 通用对象模型中的完全中止和部分中止	234	文献注释	327
11.7 带有事务中止的对象模型调度协议	237	第15章 恢复的特别问题	328
11.8 小结	237	15.1 目标和概述	328
习题	237	15.2 索引和大对象的日志和恢复	328
文献注释	239	15.2.1 重做索引页分裂的逻辑日志条目	328
第12章 崩溃恢复: 正确性概念	241	15.2.2 大对象操作的逻辑日志条目和刷出顺序	331
12.1 目标和概述	241	15.3 事务内部保存点和嵌套事务	334
12.2 系统体系结构和接口	243	15.4 在重启过程中使用并行性	338
12.3 系统模型	244	15.5 对主存数据服务器的特殊考虑	339
12.4 正确性准则	246	15.6 数据共享机群的扩展	341
12.5 算法路线图	248	15.7 小结	344
12.6 小结	250	习题	344
习题	251	文献注释	346
文献注释	251	第16章 介质恢复	347
第13章 页模型崩溃恢复算法	252	16.1 目标和概述	347
13.1 目标和概述	252	16.2 基于日志的方法	348
13.2 基本数据结构	253		
13.3 重做胜者范型	256		
13.3.1 正常操作期间的操作	256		
13.3.2 简单的三遍扫描(三趟)算法	259		

16.2.1	正常操作期间的数据备份和归档 日志	349
16.2.2	数据库恢复算法	351
16.2.3	对平均数据丢失时间的分析	352
16.3	存储冗余	355
16.3.1	基于镜像的技术	355
16.3.2	基于纠错码的技术	357
16.4	灾难恢复	363
16.5	小结	364
	习题	364
	文献注释	365
第17章	应用恢复	366
17.1	目标和概述	366
17.2	基于队列的无状态应用	367
17.3	基于队列的有状态应用	372
17.4	基于队列的工作流	374
17.4.1	故障可恢复工作流的状态和上 下文	375
17.4.2	基于排队事务的分散工作流	376
17.5	一般的有状态应用	377
17.5.1	设计上考虑的事项	378
17.5.2	服务器应答日志算法综述	380
17.5.3	数据结构	381
17.5.4	正常操作期间的服务器日志活动	382
17.5.5	正常操作期间的客户端日志活动	384
17.5.6	日志截断	385
17.5.7	服务器重启	387
17.5.8	客户端重启	388
17.5.9	正确性推理	390
17.5.10	对于多层体系结构的适用性	393
17.6	小结	393
	习题	394
	文献注释	394

第四部分 分布式事务的协调

第18章	分布式并发控制	397
18.1	目标和概述	397
18.2	同构联邦中的并发控制	398
18.2.1	预备知识	399
18.2.2	分布式2PL	400
18.2.3	分布式TO	401
18.2.4	分布式SGT	402
18.2.5	乐观协议	403
18.3	分布式死锁检测	404
18.4	异构联邦中的可串行性	406
18.4.1	全局历史	407

18.4.2	全局可串行性	408
18.4.3	准可串行性	410
18.5	通过本地的保证获得全局可串行性	411
18.5.1	严厉性	411
18.5.2	提交排序	412
18.6	基于ticket的并发控制	413
18.6.1	强迫冲突的显式ticket	413
18.6.2	隐式ticket	415
18.6.3	显示和隐式ticket的结合	415
18.7	异构联邦中对对象模式的并发控制	416
18.8	数据共享系统的一致性和并发控制	417
18.9	小结	420
	习题	421
	文献注释	422
第19章	分布式事务恢复	424
19.1	目标和概述	424
19.2	基本的两阶段提交算法	425
19.2.1	2PC协议	425
19.2.2	重启和终止协议	430
19.2.3	独立恢复	435
19.3	事务树两阶段提交算法	436
19.4	分布式提交的优化算法	439
19.4.1	假设中止协议和假设提交协议	439
19.4.2	只读子树的优化	443
19.4.3	协调者转移	445
19.4.4	减少阻塞	447
19.5	小结	448
	习题	449
	文献注释	450

第五部分 应用与未来前景

第20章	下一步是什么	453
20.1	目标和概述	453
20.2	我们完成了什么	453
20.2.1	开发者可用的解决方案	453
20.2.2	高级的系统搭建者可用的最新 技术	454
20.2.3	研究人员的方法学 and 新的挑战	455
20.3	用于普遍访问的数据复制	455
20.4	电子服务和 workflows	457
20.5	性能和可用性保证	460
	文献注释	462
	参考文献	464

第一部分 背景与动机

第 1 章

概 述

If I had had more time, I could have written you a shorter letter.

—Blaise Pascal

There are two mistakes one can make along the road to truth—not going all the way, and not starting.

—Buddha

1.1 目标和概述

事务处理是数据库与信息系统中一个重要的主题。而且，事务处理在它最初发展的领域之外正迅速获得重要的应用。在本章中，我们将讨论为何事务是一个有创意的想法，为何事务对现实生活中某种类型的数据管理及其相关问题来讲是一个合理的抽象概念，以及应用事务概念可以做什么，不可以做什么。

事务概念最初是在数据库管理系统领域发展起来的，是一种对共享数据库进行并发访问和错误处理的范型。因此，我们从讲述事务在其中起重要作用的数据库和其他信息系统的典型应用场景开始（见 1.2 节）。最初的也是最具有代表性的应用例子是银行业的资金转账；服务业目前出现了许多在功能和结构上类似的应用，尤其值得一提的是旅游业中的机票、汽车、酒店预订服务。所有这些典型的应用例子通常被称为联机事务处理（online transaction processing），简称 OLTP。另外，我们还会看到，事务的应用涉及现代的商业领域，例如电子商务和工作流管理（也称为业务处理）。

从计算机和网络的底层体系结构来讲，我们面对的将是规模可能非常庞大的分布式系统，而且系统可能是由异构的、互操作的组件组成的。最常见的情况下，这种组件是一个数据库管理系统，更确切的说，可以是一个处理客户端（工作站、个人计算机、笔记本电脑、PDA、电子传感器以及其他嵌入式系统）请求的数据库服务器（database server）。在目前信息技术多样化的情况下，电子邮件服务器、基于 Web 或局域网的文件服务器、工作流管理系统等也扮演了越来越重要的角色，而且也需要事务的支持。

事务概念所要彻底解决的最主要问题是，在高度并发的数据访问中保证数据的一致性，而不管是发生什么样的错误，这通常是一个微妙而又困难的问题。事务的另外一个重要特征是它是以对应用逻辑（以及应用开发）透明的方式实现的，因此应用程序开发人员根本不需要考虑这些问题，这就是为何事务是一个抽象概念（abstract concept），为何这个概念是现代信息技术奠基石的原因。1.3 节将从战略的视角讨论当代信息系统中事务概念的作用。我们将介绍一个通用的参考架构，作为实现和部署一个信息系统所必需的完整架构的概览；我们还将讨论几个这种架构的变体，它们在现实世界得到了广泛的应用。我们尤其还要识别在面向事务访问的体制下管理持久数据的组件，也就是事务（数据）服务器（transactional (data) server）。在 1.4 节中，我们将讨论事务概念的抽象特征，以及在事务服务器中这些特征所带来的极大好处。我们还将概要地给出对服务器算法在可靠性（reliability）、可用性（availability）以及正确性（correctness）和性能（performance）上的要求。

到目前为止，对事务数据服务器来说，最重要的实例是数据库系统。然而，本书不是一本关于数据库系统的书籍，我们所要讨论的问题都是直接与事务紧密相关的，而不涉及其他问题。我们将在1.5节概要介绍一下本书读者所应该具有的数据库系统知识。这将为我们在下一章介绍事务服务器的两个计算模型做铺垫。

像后续章节一样，1.6节对本章进行了总结，告诉读者通过对本章的阅读应该掌握哪些关键知识。

1.2 应用举例

我们通过列举几个典型的应用来开始我们的讲述，它们可以让事务的特征变得浅显易懂。这几个应用场景是：

- 银行系统的资金转账，一个典型的OLTP应用。
- 基于Web的电子商务（e-Commerce）。
- 旅行计划，一个工作流的例子。

1.2.1 联机事务处理：借/贷的例子

考虑银行系统中的一个简化的例子，银行利用关系数据库跟踪账户交易信息。数据库中有一个名为账户（Account）的表，表中用于描述账户信息的字段包括账户标识（accountID）、户主名（customer name）、分支机构标识（branchID）、余额（balance）。银行中的事务或者为取款（withdrawal）或者为存款（deposit）（这也就是为什么银行中的应用经常被认为具有借/贷事务的特征），而这些事务经常在资金转账中被合并在一起。典型的借/贷结构程序如下所示，这个程序通过把标准数据库查询语言SQL嵌入到C程序中实现。注意区分被调用程序的本地变量与被所有程序共享的底层数据库中的数据的区别。也要注意到一个现实生活中的、完备的借/贷程序可能在SQL选择（Select）语句与随后的更新（Update）语句之间包含多种对账户数据健全性的检测（例如高额取款时）。

```
/* 借/贷程序 */
void main()
{
    EXEC SQL BEGIN DECLARE SECTION;
    int accountid, amount; /* 输入变量 */
    int balance; /* 中间变量 */
    EXEC SQL END DECLARE SECTION;
/* 读取用户输入 */
    printf("Enter Account ID,Amount
    for deposit (positive) or withdrawal (negative):");
    scanf("%d%d", &accountid,&amount);
/* 获取账户的当前余额，把余额读入程序本地的中间变量 */
    EXEC SQL Select Account_Balance Into :balance
    From Account
    Where Account_Id = :accountid;
/* 金额加入余额（金额为负代表提款）*/
    balance = balance + amount;
/* 更新数据库中的账户余额 */
    EXEC SQL Update Account
    Set Account_Balance = balance
    Where Account_Id = :accountid;
    EXEC SQL Commit Work;
}
```