数据库互操作系统 正确性 安全性

计算机科学1996Vol.23№5

TP316

数据库互操作的正确性与安全性*)

朱 浩 高洪奎 吴泉源
(国防科技大学计算机系 长沙 410073)

摘 要 In this paper, we discuss both the correctness and the safety questions of the interoperation among databases. We list the various incorrect and unsafe phenomena of interoperation and analyze why these incorrect and unsafe arise. We think the correctness and the safety are important aspects of software interoperation in a distributed environment.

关键词 Database interoperation, Heterogeneous databases, Correctness, Safety.

随着网络技术的成熟与信息时代的到来,大范围内的数据共享与分布式事务成为计算机应用的迫切需求.如何使现有的数据库系统能够进行相互交互,在保持局部自治性和安全性的前提下,为用户透明地提供大范围内的数据资源共享与数据操纵能力,成为目前数据库技术研究的一个重要课题.分布异构数据库互操作系统正是在这种背景下产生的.

自80年代中期以来,对数据库互操作系统的研究在理论与工程上都开展了相当的工作,在研究路线与实现技术上都表现出多样性,包括是否构造全局模式及使用何种全局模式,是否使用全局语言及全局语言的定义,全局事务的调度等.但另一方面,对互操作的正确性与安全性等问题尚未予以关注.我们认为,数据库互操作系统是一个比较典型的分布环境下软件互操作系统,在工程上克服各种异构性实现互操作的同时,还要考虑与处理集中式环境下软件所不曾遇到的各种问题,如正确性与安全性.对于数据库的互操作,这些问题非常重要,因为数据是原始的历史记载,对安全性有严格要求;同时,由于语义的领域相关性,互操作的正确性亦是一个复杂的问题;加之分布式环境固有的各种不确定因素,使得数据库互操作的正确性与安全性成为一个值得重视的问题.

本文根据我们对数据库互操作系统的研究与实践,讨论互操作的正确性和安全性.我们的工作尚是初步的,但希望该问题能引起较广泛的重视与讨论.

为了讨论方便,这里给出一个带模式异构的分布式股票数据库样本,由三个局部数据库组成,分别位于NewYork、London和Tokyo,我们用地名作为数据库名.

Database NewYork

(仅由一个关系NS组成,每种股票一条记录)

NS:	date	stock	price
	910408	IBM	347
	910408	HP	418
	910408	GM	250
	910409	IBM	350
	910409	HP	420
	910409	GM	215

Database London

(仅由一个关系LS组成,每天一条记录,每种股票一个属性)

LS:	date	HP	IBM	GM
	910408	418	365	250
	910409	420	350	200

Database Tokyo

(每种股票一个关系,每天一条记录)

HP:date	price	IBM:date	price	GM:date	price
910408	425	910408	347	910408	385
910409	420	910409	350	910409	320

*)863资助项目(863-306-61-04)

显然，数据库 NewYork 关系 NS 中属性 stock 的值对应于数据库 London 关系 LS 中属性的名，对应于数据库 Tokyo 中关系的名，即数据库 NewYork 中的数据对应于数据库 London 和数据库 Tokyo 中的元数据(metadata)，这就是相同功能数据库中的模式异构。

一、数据库互操作的正确性

1 全局事务结果的不确定性

一个数据库互操作的全局事务结果是不确定的，这是单一数据库系统及分布式数据库系统所不具有的问题。在分布式数据库系统中，尽管数据在物理上是分布的，也存在着网络传输延时的不确定性，但不存在本地事务，所有事务都是全局的，由 DDBMS 统一管理，事务在各结点的提交与执行可通过2PC 协议予以确定化。造成互操作系统中这一问题的本质是局部事务的存在及局部数据库系统的自治性。互操作系统由于对透明性的要求及对局部自治性的维护，使得全局事务在优先权上低于局部事务(局部事务的不可中断性)。在一个全局事务从提交到在各个结点处于就绪状态，各个结点的数据库状态可能无法保持不变。以股票数据库互操作系统为例，为突出问题本质，我们假定网络传输是瞬间完成的。假定一用户在标准时间10:00AM 在数据库 NewYork 结点提交一全局事务，查询三个数据库中 IBM 公司的股票值。该事务一种可能的执行情形是，NewYork 结点的本地事务管理器(LTM)当前空闲，子事务可立即提交，10:00AM 处于就绪状态；London 结点的子事务10:10AM 处于就绪状态；而 Tokyo 结点的子事务由于本地事务繁忙，10:30AM 才进入就绪状态。这样，该全局事务读到的是 NewYork 数据库在10:00AM 的状态，London 数据库在10:10AM 的状态，Tokyo 数据库在10:30AM 的状态。该事务另一种可能的情形是，NewYork 结点子事务10:30AM 进入就绪状态；London 结点子事务10:00AM 进入就绪状态；Tokyo 结点子事务10:10AM 进入就绪状态。因为股票行情瞬息万变，在10:00AM 到10:30AM 这段时间中各地的本地数据库可能经历了若干本地事务操作而使得全局视图下的数据状态发生了变化，使事务读到的是另一番数据了，这就是全局事务结果的不确定性。通常，我们把实验结果的可重复性作为实验正确性与可靠性的一条标准，从这一标准上说，数据库互操作系统的正确性存在着缺陷。

鉴于问题产生的原因，我们不可能达到完全确定的全局事务操作，而只能使问题得到有限的解决，使用超时(timeout)是一办法。用户在提交全局事务时，给出一个超时值，在这个超时时限内事务得到执行，则假定该事务结果的正确性是可容忍的。对数据状态变化快的数据库，超时可定义得小些，对数据状态变化较慢的数据库，则可大些。

2 全局事务结果的不完备性

全局事务结果的不完备性是指全局事务未能对所有符合其操作要求的数据对象施行。造成这种操作的不完备性可有三方面的原因。

①表。对一个省缺表的全局事务，若某个应操作表所在结点机未启动，这时是拒绝该事务还是接受该事务？若拒绝该事务，则当互操作系统较大，网络结点机较多时，全局事务的接受率可能大为降低；若接受该事务，则事务的结果是不完备的。对于多数据库查询操作，这一问题尚不突出，对于多数据库更新操作，这种不完备性可能造成较严重的问题。

②域。经常，用户使用 select * from(表名) where(条件表达式)这样的查询命令。在互操作系统中，由于图式异构的存在，* 号(或 all)在各个表中并不等价，某些表中 * 号所指称的域可能比另一些表中多或少，因而 select * 所获得的查询结果相对于用户自身表中 * 的语义来可能要少，

③异名同义的属性。当存在异名同义的属性时，以该域数据为操作对象的事务同样是不完备的，且这种不完备性更为隐蔽。当远程用户知道这种情况的存在时，可用另一远程事务来补偿，当远程用户不知道这种情况的存在时，这种不完备性持续存在。

④数据共享与授权。某些系统提供了让本地数据库拥有者有选择地公布他的可共享数据并对不同用户的操作权限施以限制的机制(我们研制的 InterDB 便是如此)。毫无疑问，数据库拥有者对数据共享与授权的限制也影响了全局事务结果的完备性。对这种情况无法采取什么措施，因为它是为了维护数据私有性和安全性而必须作出的牺牲。

3 属性语义差异导致的不正确性

由于局部数据库设计的自治性及数据语义的隐

含性，使数据库互操作系统中 LDBS 间极易存在语义差异，语义差异有二类：

①同名异义的属性，如股票数据库中的 price 可以是开盘价、收盘价、最高价及最低价，不同局部数据库的设计者可能使用不同语义的 price，造成同名异义。

②异名同义的属性，如最高价可能分别用 max-point 和 highestprice 表示。

属性语义差异导致的不正确性可有下面几类：A)当使用异名同义的属性作为操作的对象时，结果是不完备的；B)当使用异名同义的属性作为操作的条件时，条件是不完备的；C)当使用同名异义的属性作为操作的对象时，得到错误的结果；D)当使用同名异义的属性作为操作的条件时，条件是错误的，整个事务都是错误而荒谬的，后三类错误都是严重的。因此，尽管语义差异是分布式系统中广泛存在而又难以克服的困难，但仍须致力于寻找有效的解决方案。在数据库互操作系统中，建立与使用全局视图 GRR 及显式表述各数据库中每个属性的语义是十分必要和重要的，在尚未找到与建立一个形式化的完备的语义描述方法之前，我们只能采用说明性的描述方法并通过人工的介入来保证语义的正确与一致。

4 命令语义差别所致的不正确性

因为全域表示符 * 号(或 all)在各个表中可能不等价，以 * 号提交的全局命令在不同的表中所施的操作亦不等价。当施行的是更新命令时，还引起不安全性。

5 与本地事务并发不当导致的不正确性

并发操作不当可导致死锁，亦可造成不正确的操作结果。当全局事务与本地事务非互斥地对同一数据对象施行操作且本地事务为写操作时，全局事务或可能读到逻辑上不一致或不完全的数据，或对逻辑上处于不一致状态的数据施行了写。

6 补偿性事务的正确性问题

补偿性事务是为了保证事务的原子性而引入的事务概念。补偿性事务的引入放宽了对事务原子性的要求，使得能在一个十分不稳定的分布环境下达到事务的某种可满意的原子性。它可有不同的所指，一种语义是撤除已提交子事务所施行的操作，使整个子事务回到未执行状态；另一种语义是对可推迟执行的子事务，反复执行直至成功；再一种语义是，在收到 commit 时由于结点机关机而未能提交的子事务，在结点机开机后首先提交执行。我们的工作采用了第三种含义的补偿事务。假如在结点机恢复后可保证待补偿的全局子事务首先得到执行，则不存在正确性问题，全局事务的原子性是严格保证的；若补偿事务不能得到首先执行，则执行时的状态可能不等价于重启动后的初始状态而引起正确性问题。前二种语义所达到的都是削弱的原子性，在某些应用情形下可能不引起正确性问题，如订票系统的订票操作可通过补偿事务撤除订票，订票的操作是可重试的(假定不指定座号)。但在另一些情形，这种削弱的原子性显然使得事务执行的正确性受到影响，对于数据及数据变化敏感的应用系统便是如此。例如，指定座号的订票操作是不可重试的。

7. 冗余数据及其不一致性

全局视图下的冗余数据是指一个对象的数据在多个局部数据库中存储与处理。例如，对一个分布式气象资料数据库，A 地气象台根据本地观测得到 A 地附近地区的气象资料，其中包含 C 地的数据；B 地气象台根据本地观测得到 B 地附近地区的气象资料，其中亦包含 C 地的数据。由于观测地点等因素的不同，C 地的气象数据在 A、B 二个数据库中可能不一致，从而造成全局视图下的冗余数据及数据的不一致。

二、数据库互操作的安全性

数据库互操作的安全性包括多个方面，主要有访问控制安全、数据传输安全、全局事务调度安全及 LDBS 数据一致性与完整性安全等。前二个问题在分布式系统中具有普遍性，在此我们只讨论分布异构数据库互操作系统所特有的全局事务调度安全及全局事务与局部事务耦合的安全。

1 全局数据不一致与不完整

全局事务的非原子性执行可导致全局视图下数据不一致与不完整，这种不一致与不完整并没有破坏局部数据库的数据一致性和完整性，而是全局视图下的不一致与不完整，并且不可能通过对局部数据库的一致性和完整性检查而得到发现。例如，在一个分布异构工资管理系统中，一个全局事务对每个结点数据库中每个人的年薪增加20%，若该事务在

某些结点失败，则造成全局数据不一致性，而这时每个局部数据库的数据都是一致性的，因而全局事务调度要求严格保证全局事务的原子性。

2 本地数据一致性与完整性的破坏

①全局事务与本地事务并发不当。这可造成本地数据库的不一致与不完整，因而全局子事务与本地事务间必须使用可靠的互斥机制来保证数据的一致性和完整性。

②部分数据共享与授权。部分数据共享可能引起本地数据库的不一致与不完整。当一个对象的数据部分处于共享表中而部分处于非共享表中时，对共享表的写、删、改操作显然将破坏数据的一致性与完整性。这种原因引起的数据一致性与完整性问题应由本地数据库拥有者负责，谨慎处理表的共享与授权，及时检查数据的一致性与完整性。

3 死锁

在数据库互操作系统中，当二个或多个全局事务僵持地相互等待时，就发生了死锁。这种等待关系可以是直接的，即仅由全局事务的子事务形成；也可以是间接的，当等待关系中亦包含了本地事务。我们说在一个局部系统 S 上一个子全局事务 gi,s 直接等待另一个子全局事务 gj,s，当且仅当 gj,s 锁了一个数据对象而 gi,s 正等待 gj,s 释放该锁。一个子全局事务 gi,s 间接等待另一个子全局事务 gj,s 当且仅当在某个结点 S 存在一个非空的本地事务集合 L1,…,Ln，L1直接等待 gj,s，Li+1直接等待 Li(0<i<n)，而 gi,s 直接等待 Ln。

如果存在一个全局事务集合 G1,…,Gn 和一个结点集合 S1,…,Sn，且出现如下的等待情形：

①G2在 S1直接或间接等待 G1，Gi+1在 Si 直接或间接等待 Gi，G1在 Sn 等待 Gn（如果在结点 S1 有 gj,s1等待 gk,s1，我们说 Gj 等待 Gk）；

②gi,si(i=1…n)处于预备提交状态，但直到剩余子事务都进入预备提交状态后才可提交；

③一个处于预备提交状态的子事务不释放任何资源，直至其提交或流产。

那么存在一个全局死锁。一个最简单的仅由全局子事务构成的全局死锁如图1所示，一个由本地事务参与构成的全局死锁如图2所示。

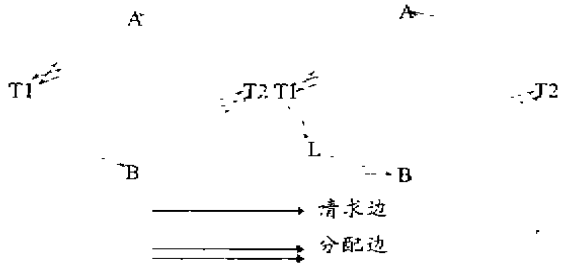

图 1　　图 2

仅由全局事务构成的全局死锁可用一般的分布式数据库系统中的死锁处理方法予以解决，而一个由本地事务参与构成的全局死锁则不易解决，因为互操作系统对于 LDBS 的局部调度毫无所知。

结束语　本文我们讨论了数据库互操作系统中可能存在的各种正确性与安全性问题。有些不正确性与不安全性可通过全局事务的构造与全局事务的调度得到解决，而另一些则是系统所固有的，如全局事务结果的不确定性。对数据库互操作正确性与安全性的全面认识有助于我们在研究开发中充分处理这些问题，且是正确安全使用互操作系统的前提。

参考文献

[1]唐雪飞等，异种数据库互操作性：概念及面临的问题，《计算机科学》1994Vol. 21 No. 6

[2]W. Litwin, et al., Interoperability of Multiple Autonoumous Database, ACM Computing Survey, 1990, Vol. 22, No. 3

[3]S. Mehrotra, et al., Ensuring Transaction Atomicity in Multidatabase Systems. In Proc. of The 11th ACM SIGACT-SIGMOD-SIGART Symp. on Principles of Database Systems, San Diego, 1992

[4]D. Georgakopolous, et al., On Serializability of Multidatabase Transactions Through Forced Local Conflicts. In Proceedings of The 7th Intl. Conf. on Data Engineering, Kobe, Japan, 1991

[5]A. Bouguettaya et al., Large Multidatabases: Issues and Directions, In Interoperable Database Systems (DS-5), Elsevier Science Publishers, 1993

[6]G. Dong et al., Representation and Translation of Queries in Heterogeneous Databases with Schematic Discrepancies, 同[5]

[7]A. Sheth et al., So Far (Schematically) Yet So Near (Semantically), 同[5]