隐式授权动态推导模型研究

Research on the Dynamic Deducing Model of Implicit Authorization

张 召 黄国兴 鲍 钰

(华东师范大学计算机系 上海200062)

Abstract With the concepts of the authorization model of OODM, this paper proposes the deducing implicit authorization model in relational database management system (RDBMS). This model helps the user gain implicit authorization by deducing the explicit authorization on the application layer. Finally the authors describe the model by predicate logic in order to provide the foundation for application.

Keywords Authorization, Implicit authorization

1. 引言

随着数据库技术、通讯技术的不断发展,信息管理系统得到了越来越广泛的应用。然而,对于任何一个信息管理系统,尤其是多用户信息管理系统,健壮有效的安全管理机制是合法使用信息,防止非法获得信息或破坏信息的基本保障。一个信息管理系统的安全管理包含系统级的安全管理和应用级的安全管理。本文讨论的授权模型在应用级别上实现了从显式授权到隐式授权的动态推导。这种授权的动态推导,对于一个有许多用户的大型信息管理系统非常有用。它不仅节约了大量的存储空间,而且也使得授权管理变得相对容易。

2. 基本概念

授权 被定义为一个三元组(s,o,a),用来表示主体 s 在对象 o 上有权力 a。

显式授权和隐式授权 如果原始事实库中存在三元组(s,o,a),则说明主体 s 在对象 o 上有显式授权 a。如果原始事实库中不存在三元组(s,o,a),但在原始事实库的基础上,由规则可推导出三元组(s,o,a),则说明主体 s 在对象 o 上有隐式授权 a。

正授权和负授权 正授权用来指定某个主体在某个对象上拥有的权力,负授权用来指定某个主体在某个对象上没有的权力。

3. 数据库授权系统中隐式授权推导模型的建立

3.1 授权的形式化定义

定义3.1 设数据库系统所有用户构成权主集 S,可标识的对象集构成权体集 O,系统设定的权型集为 A,$S\cap O\cap A=\Phi$,授权为一三元组(s,o,a),其中,$s\in S, o\in O, a\in A$,而对任一授权(s,o,a),授权测试函数 f 为映射:

$f: S\times O\times A\rightarrow (T,F,U)$

记作 f(s,o,a),其中,T=True,F=False,U=Undecided

据定义3.1给定一个三元组 (s,o,a),如果 f(s,o,a)=T 则权主 s 在权体 o 上有授权类型 a。

然而,在现实世界中,f 的值有可能沿着域 S,O 和 A 蕴涵其它的值(对于授权中不满足 f 的蕴涵规则的特殊授权,可以引入弱授权的授权覆盖来解决)。例如,在某个类 (Class)上拥有读授权的用户应能读取该类的所有实例 (Instance),即授权基于域 O 被蕴涵;一个上级用户应能享有其下属拥有的授权,即授权基于域 S 被蕴涵;又,在某个对象上拥有写授权的用户理所当然对该对象拥有读授权,即授权基于域 A 被蕴涵;所以,从某些显式的 $f(s_1,o_1,o_1)$ 中推出 $f(s_2,o_2,a_2)$ 的值,既可以排除存储 $S\times O\times A$ 中所有点的 f 的值,又能够检测不同 f 的值之间可能出现的冲突。

定义 3.1从静态(结构)角度给出了授权的形式化定义,根据该定义,我们可以建立数据库授权系统中隐式授权的推导模型。

3.2 授权主体(S)

在大型的多用户信息管理系统中,为了减少权主集的数量,将有权进入系统的用户按照拥有职权的不同划分成不同的角色,这些不同角色之间按相互之间的逻辑关系构成一棵有根有向的树,我们称为授权角色树 (Authorization Role Tree,ART)。授权角色之间满足一定的关系,可用图1表示。由图可看出不同角色之间的偏序关系为:

$S>R_k>R_{kk}>\cdots>R_{kkk}\cdots$ 其中 len(kk…)=ART 的深度。

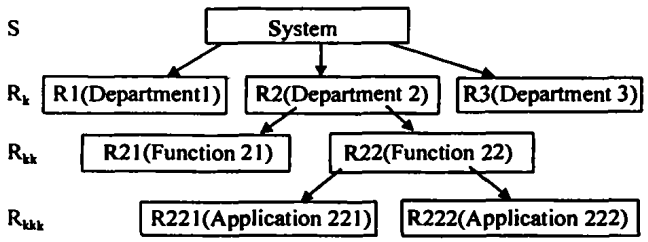

图1 授权角色树(ART)

图1中,每个角色属于一个父角色,但可拥有多个子角色。一个父角色享有其所有子角色上的全部授权。由此有如下定义和规则:

定义3.2 对于偏序关系 $s_i>s_j(s_i\in S, s_j\in S)$,表示在 ART 中有一条从 s_i 到 s_j 的蕴涵链;偏序关系 $s_i\geqslant s_j$,表示 $s_i=s_j$,或者 $s_i>s_j$,若 $s_i>s_j$,则存在有角色 $s_1,s_2,\cdots,s_n\in S$,且有 $s_i>s_1>s_2>\cdots>s_n>s_j$。

张 召 硕士研究生,研究方向:智能信息系统,数据挖掘;黄国兴 教授,研究方向:智能信息系统,数据挖掘;鲍 钰 硕士研究生,研究方向:智能信息系统,数据挖掘。

由定义3.2可知，对于ART中直接或间接连接的角色之间的蕴涵授权，如下规则和推论成立：

规则3.1 对于任意 $o\in O$ 和 $a\in A$，如果 $s_i\geqslant s_j$，则 $(s_j,o,a)\rightarrow(s_i,o,a)$。

推论3.1 对于任意 $o\in O$ 和 $a\in A$，如果 $s_i\geqslant s_j$，则 $(s_i,o,-a)\rightarrow(s_j,o,-a)$。

推论3.1从规则3.1中导出，用于负授权。显然这种推导完全遵守谓词演算规则：

$(p\rightarrow p')\longleftrightarrow(-p'\rightarrow -p)$

对规则3.1可陈述如下：如果某个角色 s_j 在对象 o 上具有授权 a，那么在ART中所有位于 s_j 上层的角色 s_i（即 $s_i\geqslant s_j$）都在对象 o 上拥有授权 a，而推论3.1则说明，如果角色 s_i 在对象 o 上有负授权 $-a$，那么ART中所有位于 s_i 之下的角色 s_j（$s_j\leqslant s_i$）在对象 o 上有负授权 $-a$。

3.3 授权类型(A)

所有授权类型按照它们之间的逻辑关系可以组成一棵有根有向树，叫做授权类型树（Authorization Type Tree，ATT），考虑到关系型数据库系统中的主要基本操作，我们在授权系统中定义如下权型：

$A=\{O,WG,RG,W,R,D\}$

相应的ATT如图2所示，其中：O—Own(所有)；WG—Write Grant(写授权)；W—Write(写)；RG—Read Grant(读授权)；R—Read(读)；D—Define(定义)。

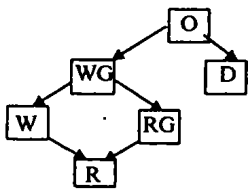

图2 授权类型树(ATT)

在图2中，一个结点代表一个授权类型，从某一结点A到另一结点B的有向边表示授权类型A蕴涵了授权类型B，因而图2就直观地反应了该授权系统中权型集的蕴涵关系，并由此可得如下定义和规则：

定义3.3 偏序关系 $a_i>a_j(a_i\in A,a_j\in A)$，表示在ATT中存在着一条从 a_i 到 a_j 的蕴涵链；偏序关系 $a_i\geqslant a_j$ 表示或者 $a_i=a_j$，或者 $a_i>a_j$，如果 $a_i>a_j$，则有 $a_1,\cdots,a_n\in A$，且 $a_i>a_1>,\cdots>a_n>a_j$。

规则3.2 对于任意 $s\in S,o\in O$，如果 $a_i\geqslant a_j$，则 $(s,o,a_i)\rightarrow(s,o,a_j)$

推论3.2 对于任意 $s\in S,o\in O$，如果 $a_i\geqslant a_j$，则 $(s,o,-a_j)\rightarrow(s,o,-a_i)$

规则3.2说明了，如果角色 s 在对象 o 上拥有授权 a_i，那么角色 s 在对象 o 上拥有在ATT中所有位于 a_i 下层的授权 a_j（即 $a_j\leqslant a_i$）。而推论3.2则说明，如果角色 s 在对象 o 上拥有负授权 $-a_j$，那么角色 s 在对象 o 上拥有在ATT中所有位于 a_j 上层的 a_i（即 $a_i\geqslant a_j$）的负授权 $-a_i$。

3.4 授权对象(O)

数据安全的目标，应是有选择地使用公共数据，把用户权限分级，限制对数据的存取和操纵，因此，要求数据的组织有一定的粒度，并能加以控制。按照这一原则，授权系统的授权对象集合可分解为与不同角色有关的多个逻辑保护对象，一个应用系统中，不同的逻辑保护对象之间的逻辑关系就可以构成一棵有根有向的授权对象树（Authorization Object Tree，AOT），如图3所示。

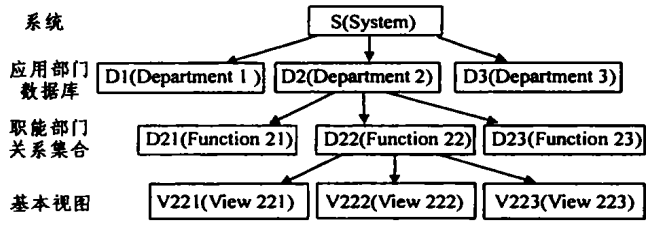

图3 授权对象树(AOT)

一个AOT是一棵向下生长的有根有向树，它的每一个结点都是一个可独立授权的对象，AOT中的一条有向边称之为蕴涵链，它定义了两个结点上所代表的一对授权对象之间的蕴涵关系，从某一节点A到另一节点B的边表示对A对象上的授权蕴涵了对B对象上的授权，由此我们得到如下的定义和规则：

定义3.4 偏序关系 $o_i>o_j(o_i\in O,o_j\in O)$，表示在AOT中从 o_i 到 o_j 存在一条蕴涵链；而 $o_i\geqslant o_j$ 则表示，或者 $o_i=o_j$，或者 $o_i>o_j$，如果 $o_i>o_j$，则有 $o_1,\cdots,o_n\in O$，且 $o_i>o_1>\cdots>o_n>o_j$。

在两授权对象之间蕴涵链上的蕴涵授权遵守如下规则和推论：

规则3.3 对任意 $s\in S,a\in A$，如果 $o_i\geqslant o_j$，则 $(s,o_i,a)\rightarrow(s,o_j,a)$

推论3.3 对任意 $s\in S,a\in A$，如果 $o_i\geqslant o_j$，则 $(s,o_j,-a)\rightarrow(s,o_i,-a)$

规则3.3说明了，如果角色 s 在对象 o_i 上拥有授权 a，那么角色 s 在AOT中所有位于对象 o_i 下层的对象 $o_j(o_j\leqslant o_i)$ 上拥有授权 a。而推论3.3说明，如果角色 s 在对象 o_j 上拥有负授权 $-a$，那么角色 s 在AOT中所有位于 o_j 上层的对象 o_i（即 $o_i\geqslant o_j$）上拥有负授权 $-a$。

4. 隐式授权推导模型的形式化表示

为节省存储空间，初始时给出的最小显式授权集，满足如下条件：

$\neg(\exists\ (X,Y,Z)(abop(X,Y,Z)\rightarrow abop(S,O,A)))$

上式中论域为显式授权集，abop表示授权操作符。

为使授权简洁明了，设在该授权模型中不存在不确定性，一个授权主体 s 要么对授权对象 o 有正授权 a，要么有负授权 $-a$，不存在不确定状态。

4.1 谓词说明

(1)用一个三元谓词 abop(X,Y,Z)来表示授权的语义，其中第一个参数表示授权主体，第二个参数表示授权对象，第三个参数表示授权类型。例如，abop(s,o,a)表示授权主体 s 在授权对象 o 上有权力 a。

(2)二元谓词 rup(X,Y)用来表示系统中不同角色之间的偏序关系(上下级关系)，例如，rup(s_1,s_2)表示授权主体 s_1 在角色上是授权主体 s_2 所属角色的上级。

(3)二元谓词 oup(X,Y)用来表示授权对象集合内部元素间的组成关系，例如，oup(o_1,o_2)表示授权对象 o_2 是 o_1 的组成部分。

(4)二元谓词 aup(X,Y)用来表示授权类型集合内部元素之间的包含关系，例如，aup(a_1,a_2)表示授权类型 a_1 包含授

权类型 a_2。

(5)零元谓词 error( )用来表示授权系统中的异常。

4.2 规则库的组成

1)推导规则。根据谓词 rup(X,Y),oup(X,Y),以及 aup(X,Y)本身的传递性以及规则3.1,3.2,3.3与推论3.1,3.2,3.3可有如下九条推导规则生成。

(1)rup(X,Y) & rup(Y,Z)→ rup(X,Z)

(2)oup(X,Y)& oup(Y,Z) → oup(X,Z)

(3)aup(X,Y) & aup(Y,Z) → aup(X,Z)

(4)rup(S_i,S_j) & abop(S_j,O,A) → abop(S_i,O,A)

(5)rup(S_i,S_j) & abop(S_i,O,−A) → abop(S_j,O,−A)

(6)aup(A_i,A_j) & abop(S,O,A_i) → abop(S,O,A_j)

(7)aup(A_i,A_j) & abop(S,O,−A_j) → abop(S,O,A_i)

(8)oup(O_i,O_j) & abop(S,O_i,A) → abop(S,O_j,A)

(9)oup(O_i,O_j) & abop(S,O_j,−A) → abop(S,O_i,−A)

2)一致性检查规则。主要用来保证授权在 S,O,A 三个域上传播的正确性和有效性。

(1)角色一致性规则

·角色至多只有一个上层角色

rup(S_1,S_2) & rup(S_2,S_1) & S_1<>S_2→ error ( )

·角色层次结构上无循环

rup(R_1,R_2)&…& rup(R_i,R_{i+1}) &…& rup(R_{j-1},R_j) &… & rup(R_{n-1},R_n) & R_i=R_j & i≥1 & i≤n & j≥1& j≤n → error( )

(2)授权对象一致性规则

·授权对象集合间的无循环性

oup(O_1,O_2)&…&oup(O_i,O_{i+1}) &…& oup(O_{j-1},O_j)&… &oup(O_{n-1},O_n)& O_i=O_j& i≥1& i≤n & j≥1& j≤n → error ( )

(3)授权类型一致性规则

·授权类型集合元素间的无循环性

aup(A_1,A_2) &…&aup(A_i,A_{i+1})&…& aup(A_{j-1},A_j)& …&aup(A_{n-1},An) & A_i=A_j & i≥1 & i≤n & j≥1& j≤n → error ( )

(4)授权事实库的一致性保证规则

abop(S,O,A) & abop(S,O,−A) →error ( )

(5)授权事实库的无冗余性保证规则

→(∃ (X,Y,Z)(ab(X,Y,Z) →ab(S,O,A))))

论域为当前授权事实库。

结束语 我们提出的隐式授权推导模型适用于关系型 DBMS 环境下的多用户信息系统,在给定显式授权的情况下得到系统中所有角色的隐式授权。这种动态生成隐式授权的方法不但节约了存储空间,而且使信息系统中的授权管理变得容易和简单。但是,时空永远是一对矛盾,这种方法虽然节省了存储空间,却以增加运行时间为代价。

参 考 文 献

1 Rabbitti F,Bertino E,Kim W,Woek D. A model of authorization for next-generation database systems. ACM Transactions On Database Systems, 1991,16(1):88～131

2 Bertino E, et al. An Extended Authorization Model for Relation Databases. IEEE Transaction on Knowledge and Data Engineering,1997,9(1)

3 Bertino E,et al. Authorizations in relational database management systems. In:Proc. ACM Conf. on Computer and Communications Security,Fairfax, VA, Nov. 1993. 140～150

4 Jajodia S,Samarati P,Subrahmanian V S. A Logical Language for Expressing Authorization. IEEE Symposium on Security and Pricacy,1997

5 罗雪平. 一种扩展的基于角色的访问控制的设计与实现:[硕士学位论文]. 华东师范大学计算机系,2001

6 王景光. 数据库授权系统的研究与应用. 计算机应用,1997,5

(上接第16页)

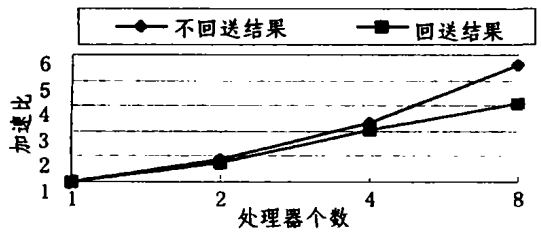

图4 数据库记录并行搜寻的加速比曲线

结论 本文针对环形 SIN 互连的并行计算机系统讨论了可分解任务的分配和调度,处理器间通过环来传递数据和消息。我们的模型同时把通信启动时间和通信延迟时间都考虑进去,能有效反映大量的现实并行计算机系统。本文重点考虑处理结果通过环形 SIN 传回源结点的情况,并为并行发送和串行发送两种情况分别建立了反映子任务处理和通信情况的线性方程组,通过对方程组进行求解可得出保证任务完成时间最短的分配和调度方案。文章给出了算法的时间复杂度和线性加速比,并进行了实验。设处理器的个数为 n,则当合理解存在时求解方程组的时间复杂度为 O(n),否则在 O(nlogn)的时间内可找到方程组的最大可用集,故本文提出的方案简单实用,能有效地在同构和异构并行系统上进行可分解任务的分配和调度。

参 考 文 献

1 Blazewicz J, Drozdowski M, Markiewicz M. Divisible Task Scheduling-Concept and Verification. Parallel Computing, 1999, 25:87～98

2 Culler D E,et al. LogP: a practical model of parallel computation. Communications of the ACM,1996, 39(11):78～85

3 Mani V, Ghose D. Distributed computation in linear networks: Closed-form solutions. IEEE Transactions on Aerospace and Electronic Systems,1994,30(2) :471～483

4 Blazewicz J,Drozdowski M. Scheduling divisible jobs with communication startup costs. Discrete Applied Mathematics ,1997,76 (1-3)

5 Bharadwaj V,Ghose D,Mani V. Optimal sequencing and arrangement in distributed single-level tree networks with communication delays. IEEE Transactions on Parallel and Distributed Systems, 1994,5(9):968～976

6 Blazewicz J,Drozdowski M. Performance limits of two-dimensional network of load-sharing processors. Foundations of Computing and Decision Sciences,1996,21(1): 3～15

7 Blazewicz J,Drozdowski M. Scheduling divisible jobs on hybercubes. Parallel Computing,1995,21:1945～1956

8 Drozdowski M,Glazek W. Scheduling divisible loads in a three-dimensional mesh of processors. Parallel Computing,1999,25:381～404