利用RBAC扩展MySQL访问控制机制

Improve the Identity Authentication of MySQL by RBAC

摘　要：MySQL是一个目前广泛用于Linux环境下的网络数据库管理系统，但其安全访问控制机制相对较弱。本文提出一个利用RBAC扩展MySQL安全访问控制功能的方法。该方法对其他环境下的数据库应用管理系统也具有普遍意义。

关键词：RBAC 访问控制 MySQL

钱　菁（华中师范大学物理系　430079）
王先荣（武汉理工大学继续教育学院　430074）

MySQL是一个多用户、多线程SQL关系数据库服务器，适用于各种流行操作系统，其主要目标是快速、健壮和易用。它是具有客户机/服务器体系结构的分布式数据库管理系统。目前MySQL以其不巧、灵活、高效、易扩展的数据管理方式在网络数据库中得到越来越广泛的使用。但其安全控制机制相对较弱。本文利用基于角色访问控制的基本思想扩展MySQL访问控制机制，从而加强其安全性。

1 MySQL的访问控制机制

MySQL提供一套简单的安全管理措施，包括内部安全性（针对服务器主机中的用户）和外部安全性（针对从网络上连接到服务器的客户机）。内部安全性涉及文件系统级的问题。外部安全性是防止MySQL服务器遭受通过网络与服务器连接请示导致的数据库内容访问攻击。本文主要针对外部安全性进行讨论。

MySQL外部安全性控制的基本作用是给某个主机上的用户对某个数据库对象权限。在网络上连接到服务器的客户机对MySQL数据库的访问由授权表user、db、host、tables_priv和columns_priv内容控制的。User表列出可连接到服务器的用户，并指定用户拥有适用于所有的数据库全局权限。Db表列出数据库以及哪些用户拥有访问这些数据库的权限，适用于数据库中所有的表。Host表与db表结合使用，在更细的级别上控制对特定主机的数据库访问权限。Tables_priv表指定表级的权限，适用于表中所有的列。Columns_priv表指定列级的权限，适用于表中特定的列。授权表中包含两种类型的列：作用域列和权限列，前者决定一个项何时可用，后者决定一个项可授予哪些权限。数据库和表的权限包括ALTER，CREATE，DELETE，DROP，INDEX，INSERT，UPDATE，REFERENCE，SELECT。管理权限包括FILE，GRANT，PROCESS，RELOAD，SHUTDOWN。

在使用MySQL时，客户机访问控制有两个阶段。第一阶段发生在连接服务器时刻。服务器查找user表看看是否能够找到与名字、正在连接的主机以及所提供的口令相匹配的项。如果不匹配，则不能连接。若匹配，则建立连接，然后继续进行第二阶段。在此阶段中，对于发布的每个查询，服务器都会依次检查各授权表以查看是否具有充足的权限来执行该查询。第二阶段继续，直到关于该服务器的会话结束为止。

从上面的介绍可知，MySQL采用一种灵活简单而非标准的访问控制方式。其控制字段除了普通数据库对象，还包括访问客户机IP。不同于目前一些主流的数据库系统采用基于角色的访问控制方式，MySQL没有角色的概念，从而使得安全访问控制弱且灵活性差，这一点对于安全管理是个缺陷。

2 RBAC及应用系统中扩展

应用级安全管理研究一直备受关注。RBAC96（Role-based Access Control基于角色的访问控制）和ACBAC97（Adminstration of RBAC基于角色的访问控制管理）授权模型是其中研究最多、思想最成熟的访问控制机制。其核心思想是将访问权限与角色相联系，通过给用户分配合适的角色，让用户与访问权限相联系。基于角色的访问控制（RBAC）结构模型见文献[1]。

MySQL自身没有角色管理功能。我们利用RBAC的基本思想，提出一种在应用程序层统一实现对整个系统的基于角色的权限管理，如图

1所示。这个模型与RBAC的不同之处在于图1的模型中角色权限分配是将应用程序层系统功能项的操作权限分配给角色，而不是将访问数据库权限分配给角色；角色通过操纵系统的功能项来操纵数据库资源，而不是通过访问数据库的权限来操纵数据库资源。

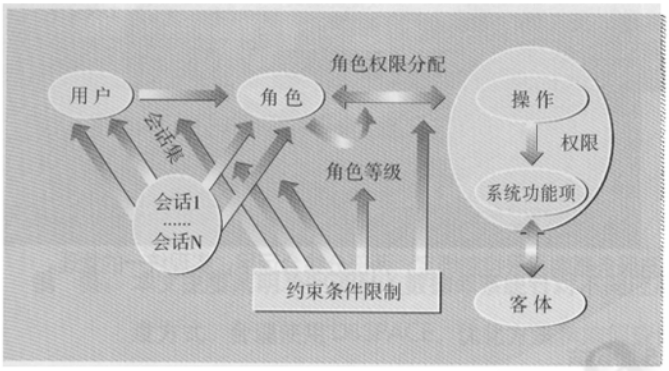


图 1　应用程序层实现基于角色的权限管理模型

对系统管理员而言，系统功能项是易见并易于理解的，它以应用、菜单、窗体、按钮的形式反映在系统应用程序界面上，而且系统功能项设置一般与应用系统的业务管理方式相适应。当角色权限改变时，只需改变角色可操作功能项，系统会自动对后台数据库相关具体数据对象访问控制权进行维护，从而使系统权限管理更加简单安全有效。

3 分析与实现

实现包含两个层次：对数据库访问权限的管理和对应用系统功能层的访问权限管理。通过综合考虑这两个层次，将前端的“功能权限控制管理”和后台的“数据存取权限管理”合为一体，进一步加强系统安全管理的性能。

依据是否具有权限管理的授权职能，将操作员用户分为两大类：普通用户和管理用户。应用系统的权限管理可看作用户对对象的操作，即<user,object,operate>这样一个三元组。对一个数据库应用系统，从最终意义上说，对象是数据库数据，用户通过操纵系统应用级模块来实现对数据库数据的操作。数据库系统对数据对象进行安全控制，是通过数据库管理员进行管理。普通用户是通过系统应用模块对数据库进行存取的用户。从系统模块执行权限上分为四层：

（1）应用执行权限；

（2）菜单执行权限；

（3）主窗口执行权限；

（4）主窗口所包含对象的执行权限。普通用户所属角色能否执行一个应用由二元组<role,application>决定，能否执行一个菜单项由三元组<user,application,menuitem>决定，能否执行一个主窗口由三元组<user,application,window,object>决定。能否执行一个主窗口所含对象项由四元组<user,application,window,object>决定。普通用户对数据库数据的操作权限分为两部分：普通用户通过执行系统模块实现的数据存取和不同操作员用户处理同一模块时支持的不同数据集。普通用户通过执行系统模块实现的数据存取是由该模块与数据库连接时使用的数据库用户决定的。不同普通用户处理同一模块时处理的数据集不同，需要对不同用户使用同一模块作某些权限控制。一般来说都采用以数据的某些特定属性取值作用户权限级别和数据库数据的元组权限级别控制实现元组层权限控制。

管理员用户在前台界面上进行相关普通用户角色的可操作功能项的设置后，后台DBMS同时进行相应数据存取权限管理，根据前端对角色功能项的指派，将有关功能项对应的数据库对象的存取权限授予角色写入数据库后台中的系统关系数据表，创建前台设置的新用户、新角色，同时完成对用户的功能角色授权，将相应功能编码数据项写入后台中程序员建立的数据库关系表中，从而实现前后台安全控制的一致性。角色名既关联数据库访问权限表，也关联功能访问权限表。当这样一个角色分配给某个用户时，此用户便拥有了该角色的权限，有效的将角色和库数据操作权限和应用功能执行权限糅合在一起，实现基于角色的有效访问。依据以上分析，我们给出如图2示用程序层实现基于角色的权限管理的数据模型。

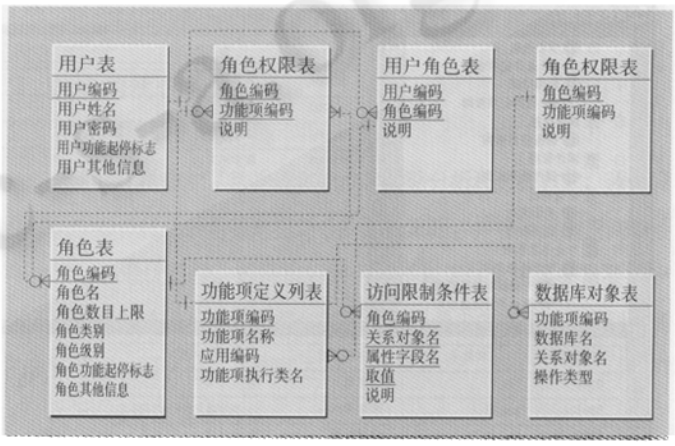


图 2　应用程序层实现基于角色权限管理的数据模型

注：功能项执行类名（classname）对应窗口，窗口内对象或菜单项等

对上面谈到的这样一种权限控制模型而言，其操作权限管理与控制是绝大多数计算机应用系统的一个子系统，其管理分为操作员、角色、存取权限、功能权限和元组权限5个层次，对保证整个系统正常安全运行有至关重要的作用。通过客户端程序进行权限管理实现前后台统一动态管理。这个权限管理模型综合考虑了数据库层和功能层权限管理两方面，在系统数据库与用户数据库间实现统一的管理方式。在

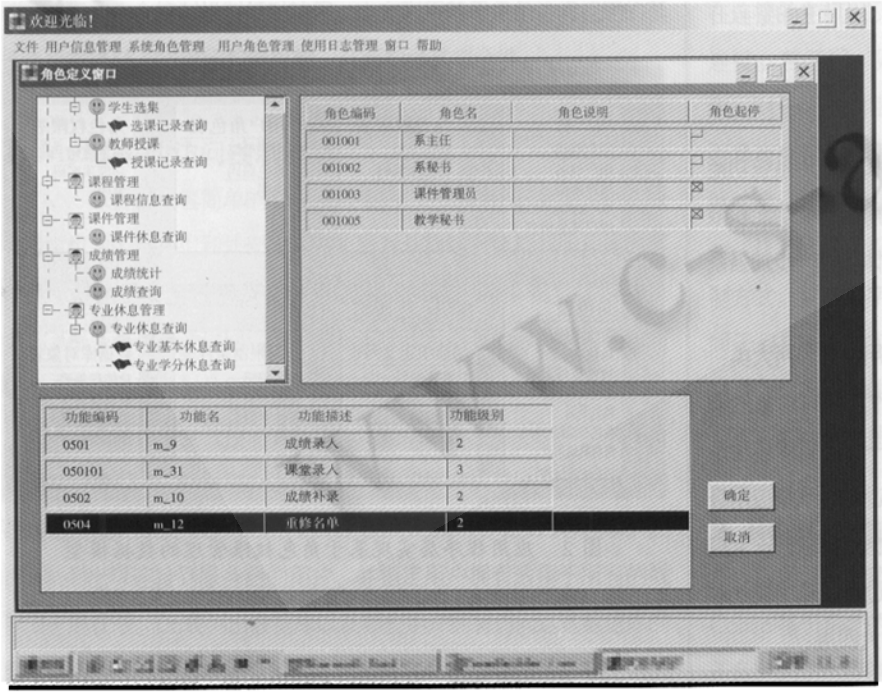
实际运用中，可依据系统具体需求，单独使用应用层的权限管理，也可两部分综合运用，实现统一的管理。从另一方面而言，并非每个系统都需要上述从应用、菜单直到每个窗口的复杂控制环节，可依据这种控制模型，视具体系统需求，灵活采用。对于单独使用功能层权限管理部分，其安全性较弱，应保证用户身份的严格鉴别和系统权限管理数据库的严格安全管理。

笔者利用上述基本思想，在基于linux的网上教学系统中，充分利用自定义角色，实现系统的权限控制。由于MySQL自身安全控制的一些限制及应用系统需求，主要是用于功能层权限管理部分。其实现如图3。

4 在B/S系统中的扩展及改进

上述权限控制方式主要针对C/S系统而言。因为C/S系统在客户端控制方法灵活，较容易实现复杂的客户端控制，大数据量的显示控制界面和较强的交互性。

而对于B/S模式的系统，一般来说在复杂的客户端控制方法上能力较弱。在B/S系统中的权限控制模型，与前面提到的C/S系统的权限控制模型从基本思想上是一致的，同样可实现前后台的统一控制和角色功能的绑定。



其不同之处在于在B/S系统中，不同用户并非是打开不同的应用窗口，而是打开不同的页面。即用户通过web server访问的软件系统是一些具有特定功能的文件集合，文件就是权限划分的最小单位。基于这样的考虑，我们把这样一个系统看作不同功能的文件集合，将文件作为权限控制的单位，把用户权限分解为某一个文件集合的可访问权。其实现数据模型如图4所示。

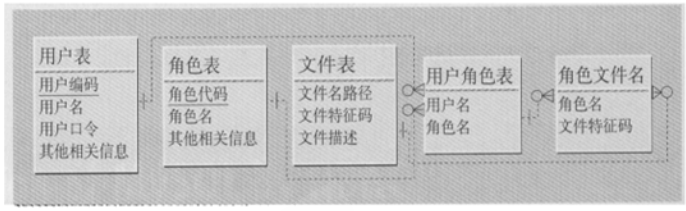


图 4 B/S 实现基于角色权限管理的数据模型

其控制模型和前面关于C/S系统类似，既可单用于客户端，也可与服务器端权限控制绑定，两部分综合运用，实现统一的管理。

5 小结

一套管理信息系统设计和开发的最后步骤就是应用系统的权限管理。本文讨论的权限控制模型对管理信息系统中的权限管理模块具有应用上的普遍性，且授权维护方便、可控性强。

对于一个数据库管理信息系统，在底层的DBMS中有基表和视图等对象，在应用层有应用模块、菜单、窗口、按钮等对象。同时角色还可划分为不同等级，各级角色间通过角色继承形成偏序关系，引入约束机制等。如何从这些对象中抽象出功能角色和抽象的粒度大小是每个具体系统要认真考察的问题，也是RBAC模型中非常重要的概念组成部分，则还有待进一步探讨。

参考文献

1 R.Sandhu,E.Coyne,H.Feinstein,C.Youmen,Role-Based Access Control Models,IEEE Computer,1996。

2 吴煲宁、孙志挥，基于角色的CIPS动态功能授权系统的设计与实现，计算机工程与应用，2001.2。

3 刘红岩、何军，PowerBuilder原理与应用指南，电子工业出版社，1999.6。

4 陈庆章、林建明，WWW与数据库集成系统的用户权限管理，计算机工程与应用，2001.6。

5 www.mysql.com mysql官方网站。