基于角色的访问控制框架的研究与实现

季永志[1,2] 阎保平[1] 续岩[1,2] 吴开超[1] 沈志宏[1]

(1 中国科学院计算机网络信息中心数据库应用研究室, 北京 100080) (2 中国科学院研究生院, 北京 100039)

摘 要: 介绍一种基于角色的访问控制框架, 详细阐述了该框架的实现原理和工作机制。基于角色的访问控制(RBAC)是一种关于授权管理的概念模型,与传统的授权策略相比,它更加灵活、安全且易维护。基于角色的访问控制框架在原理上对 RBAC 基本模型进行了概念扩展,根据大规模数据资源授权管理的需要,提出了更加细化的资源概念和明确的权限判定机制。

关键词: RBAC, 资源类, 静态资源, 动态资源, 角色包含, 三元组判定

中图法分类号: TP309.2 **文献标识码:** A **文章编号:** 1000-7180(2005)11-100-04

Research and Implementation of Role-Based Access Control Framework

JI Yong-zhi, YAN Bao-ping, XU Yan, WU Kai-chao, SHEN Zhi-hon

(1 Scientific Database Department Computer Network Information Center, Chinese Academy of Sciences, Beijing 100080)

(2 Graduate School, Chinese Academy of Sciences, Beijing 100039)

Abstract: A framework of Role-Based Access Control and its implementation mechanism and principle are introduled in detail in this paper. Role-Based Access Control (RBAC) is an abstract model on authorization management. It is more flexible, secure and easy to maintain, compared with the traditional authorization strategies. This framework of Role-Based Access Control extends the conception of the basic moddel of the RBAC. A concept on more minute granular resource and a mechanism of authorization judgment are presented, in order to fulfill the requirements of the authorization management of large-scale data resource.

Key words: RBAC, Resource class, Static resource, Dynamic resource, Role including, Judge on triple

1 引言

网络互联带来了资源共享和信息交互的便捷,随之而来的信息安全问题也越发突出,良好的访问控制策略对于保证信息的安全合理利用具有至关重要的意义。

传统的访问控制机制通常是直接为每一个用户或者一组用户赋予一系列许可权,这种做法结构复杂且缺乏灵活性,一旦企业的组织结构或系统的安全需求有所变动, 就要进行大量繁琐的授权变动,系统管理员的工作将变得非常繁重复杂,并且容易发生错误, 从而造成一些意想不到的安全漏洞。目前基于角色的访问控制已普遍应用于许多国际知名大型数据库系统 (如 Oracle、Informix、Sybase 等)的授权管理中,这些数据库系统已经实现了对 RBAC 基本模型较好的支持。DBMS 处理的资源类型相对较少,资源之间关系比较简单固定,较容易实现 RBAC 模型下的管理。但是在很多大型信息系统中,管理的往往不只是数据库相关的资源,还有很多其它类型的资源,这些大规模的数据资源类型众多而且关系复杂,要对这些资源实现基于角色的管理仅仅应用 RBAC 的基本模型是远远不够的,必须对 RBAC 的模型做概念上的扩展,结合实际的应用系统,建立 RBAC 的扩展模型。中国科学院院级重点项目"科学数据库及其应用系统"[1]就是这样一个大规模的信息系统, 要对它做复杂的授权管理, 由此提出了基于角色的访问控制框架。

收稿日期: 2005-07-28

基金项目: 中国科学院"十五"信息化建设重大资助项目 (INF105-SDB)

2 基于角色的访问控制(RBAC)

在基于角色的访问控制下,访问决策是基于角色的。角色被分配给一个组织团体内的各用户,用户通过被赋予不同的角色(比如:医生、护士、银行出纳员、经理等)而获得相应的访问许可权。

RBAC 的基本模型如图 1 所示。

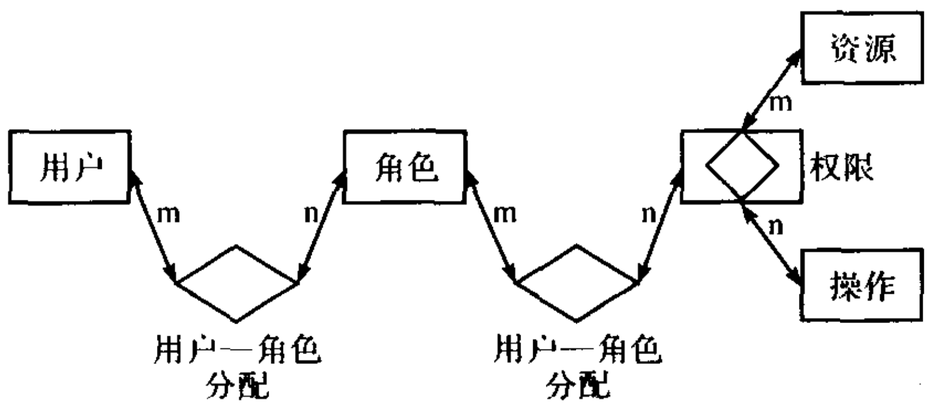

图1 RBAC基本模型图

2.1 基本概念模型

用户(User):访问系统的人。用户访问系统时可能是匿名的也可能是非匿名的。

角色(Role):由相关术语描述的工作职能或者工作名称。它表示在组织机构内将角色赋予给用户从而使该用户,具有与该角色相关的权利和责任。

权限 (Permission):对资源对象的一个或一系列的操作。

操作(Operation):对资源的动作。调用操作会引起受保护资源信息的流入或流出,或者会引起系统资源的消耗用尽。RBAC 控制的操作以及资源类型由具体实现系统来决定。比如在文件系统中的操作可以归结对文件或程序的读、写、执行等。

资源(Resource):任何一个访问控制机制的目的都是为保护信息或其它资源。在 RBAC 系统中,资源可以是操作系统中的文件、目录,或者是数据库系统中的表、行、列等包含的信息,或者是打印机、硬盘空间、CPU 时间之类可能用尽的系统资源。

资源和操作的组合构成权限,权限被分配给角色,角色被分配给用户,角色与权限之间是多对多(m:n)的对应关系,用户与角色之间也是多对多(m:n)的对应关系。用户通过作为角色成员而获得权限。角色是描述用户和权限之间多对多关系的桥梁。

2.2 分层次的 RBAC

层次(hierarchy)是定义角色之间的包含关系。一个角色的实例代表了用户个体和权限个体之间多对多的关系。在实际应用中,某个组织内可能存在一些通用的权限, 这些权限会被大量的用户使用,重复的指定这样的通用权限给角色会比较冗烦而且效率低下。为了提高效率并且支持组织化的结构,RBAC 引入了角色层次的概念。层次确定角色之间的包含关系,包含类似于面向对象系统中继承的概念。假如角色 A 包含角色 B(或者说角色 A 继承角色 B),那么角色 A 将继承角色 B 所有的特性。如果有用户 user 被直接地赋予了角色 A,由于角色 A 包含角色 B,那么他也被间接地赋予了角色 B,他可以执行所有与角色 B 相关联的权限操作。

2.3 职责分离(Separation of Duty)

职责分离包括静态职责分离和动态职责分离。

静态职责分离(SSD):不同角色之间可能存在着冲突。如果用户通过角色授权获得若干个相冲突角色所对应的权限, 那么将可能引起利益的冲突。在静态职责分离的约束下,若某个用户被授予了一个角色 A,那么他将被禁止授予其它与 A 相冲突的角色。比如一般公司里都不允许会计和出纳由同一个人担任。

动态职责分离(DSD):动态职责分离和静态职责分离一样也限制了用户的权限,在 DSD 中相冲突的角色可以赋予给同一个用户,但是在该用户的一个会话过程中不能同时扮演两个相冲突的角色。

2.4 最少权限原则

有选择地给用户分配权限,使他能够完成自己的工作任务而且又不会具有除此之外的任何多余的权限。这是一种注重实效性的管理实现。某些个体在获得权限完成他本职工作的同时,可能会利用获得的一些多余的权限去越权进行一些不必要的甚至可能是具有潜在危害的操作,最小权限原则的存在防止出现这种情况可能带来的问题。为了遵守最小权限的原则,必须仔细分析,确定工作的职能,归纳出完成每个工作所需要的权限集,以及在那个权限领域里对用户的限制。

3 基于角色的访问控制框架的实现

基于角色的访问控制框架是“科学数据库及其应用系统”中权限管理的重要组成模块,“科学数据库及其应用系统”包含众多子系统,每个子系统又包含着各自独立的数据应用模块。对于如此庞大的系统,各种资源的授权管理是件非常繁琐复杂的事情, 基于角色的访问控制框架依照扩展后的 RBAC 模型,实现了对应用系统中各种资源的授权管理。

RBAC 是一个概念上的模型, 在实际的系统实现中,必须依据模型进行实际的分析、扩展和概念划分。对于应用于科学数据库系统的基于角色的访问控制框架来说,其关键主要是如何扩展 RBAC 基本模型,描述和划分出 RBAC 模型中的五个基本元素(用户、角色、权限、资源、操作)以及它们之间的相互关系,构建出良好的权限判定机制,从而归纳出正确合理的应用模型。

3.1 分析设计

资源通常是应用相关的, 即资源属于某个应

用。科学数据库系统中存在着各种各样的资源,不同资源对应着不同的操作。对于网页(web)资源类来说,作用在其上的操作只是简单访问读取。而数据集(dataset)、数据库(datebase)、数据表(tables)虽然是不同的资源类,但是它们之间存在着一定的依附关系,并且各自对应着一系列不同的操作集合。

在这里提出了资源类的概念,即所有符合某一条件、具有某种共性的资源归为一个资源类。它是对资源特性的一种抽象,如数据表资源类 tables,它是对所有数据表资源的抽象概括。与某个资源类对应的有一个相关联的操作的集合。比如和数据表资源类 tables 对应的操作集合一般是 {Insert, Delete, Update, Select, Alter, References, Index}。对于网页资源类 web 来说,对应的相关联操作集合只有一个元素{Visit}。不同的资源类对应着不同的操作集合。当定义属于某个资源类的一个资源实例时,与该资源实例对应的操作只能是它所属资源类对应操作集合的子集。比如定义了属于数据表资源类的一个数据表资源实例表 AUTH_RESOURCE,那么与对于该数据表 AUTH_RESOURCE 的操作只能是操作集合 {Insert, Delete, Update, Select, Alter, References, Index}的子集,可能是{Select}或其它。

按照授权过程中的不同特征,实际的资源实例分为静态资源和动态资源。静态资源是指在角色定义阶段可精确解释为具体资源对象的资源。动态资源是指在角色定义阶段不能确定为具体资源对象,而需要通过资源模型描述的资源。

静态资源对应于特定类别的特定实例对象,根据不同的应用,静态资源拥有不同的属性(Attribute)。如属于数据集类的静态资源拥有属性 ID、NAME、CREATOR 等等。而属于 web 类的静态资源可能拥有属性 URL。所有静态资源都具有一个固定属性,即资源类。资源类对应于某种资源的类别,在实际存储中可能对应于一个类别代码,该代码即资源类代码(CLSID)。如数据集的资源类代码可能为 dataset,而数据表的资源类代码可能为 tables,如表 1 所示。

表 1 静态资源分配示例

PERMISSION	RESOURCE	OPERATION
Permission1	dataset:NAME= 'NANO INFO'	Read
Permission2	web:URL='http:// www.csdb.cn/ admin_view.jsp'	Visit

动态资源对应于一组同类资源的实例对象的集合,该集合可以由资源描述语言(RDL)描述。一般来说,资源描述语言会使用到资源类的 CLSID 和属性,如: web: URL like 'http://www.csdb.cn/%',该语句用以描述 URL 以http://www.csdb.cn/开头的一组 web 资源。资源描述语言也可能会使用到目前处于有效会话期的用户属性,如: dataset: CREATOR= '<USERID>'。该语句描述由具有当前角色的用户创建的数据集。用户的属性由<>括起,其内容来源于当前用户,如表 2 所示。

表 2 动态资源示例

PERMISSION	RESOURCE	OPERATION
Permission3	dataset:CREATOR= '<USERID>'	Write
Permission4	web:URL like 'http: //www.csdb.cn/%'	Visit

某些资源实例之间存在着依附关系,不同的操作之间可能会存在着包含关系。资源实例和操作组合成权限,各个权限被分配给相应的角色,角色之间也存在了包含的层次关系。若用户'jyz'以创建者的身份构建了两个数据集, 分别是 {dataset: ID= '20040601'; NAME='NANO_INFO'; CREATOR= 'jyz'} 和 {dataset: ID ='20040602'; NAME = 'NANO_PRODUCT';CREATOR='jyz'}。角色分配如表 3 所示。

表 3 角色分配示例

USEB	ROLE	PERMISSION
USERID='jyz'	NanoDatasetCreatoe	Permission3(见表 2)
USERID= 'anonymous'	AnonymousRole	Permission1(见表 1)

显然角色 NanoDatasetCreator 包含了角色 AnonymousRole,在表 3 中用户'jyz'直接的赋予了角色 NanoDatasetCreator,他同时也被间接的赋予了角色 AnonymousRole, 当 'jyz' 用户在访问数据集 {dataset: ID='20040601'; NAME='NANO_INFO'; CREATOR='jyz'}时,他既可以行使 NanoDatasetCreator 角色对该数据集的 write 操作的权利(Permission3), 同时又可以行使 AnonymousRole 角色对该数据集的 read 操作的权利(Permission1)。

权限判定机制采用三元组判定法, 即任何对某种资源进行某种操作的请求都归结为请求三元组 T=(用户,请求资源,请求操作)。判定请求三元组是否合法的操作命名为 judge(T),即判断 judeg(T)为真

(true)或假(false)。可以由 T 分解出二元组 D=(请求资源,请求操作)。通过用户→角色→权限→(资源,操作)的关联关系获得用户所具有的所有权限(资源,操作)的集合 P。有 $D \in P \Rightarrow judge(T)=true$,即如果(请求资源,请求操作)属于集合 P,则可以判定请求三元组合法,用户具有合法权限对所请求资源进行所请求的操作。否则判定请求三元组不合法,必须拒绝用户的请求。

3.2 系统实现

系统概要如图 2 所示,基于角色的访问控制框架由授权管理系统、权限检验系统以及授权相关信息数据库三部分组成,完成对应用站点上各类资源的访问控制。

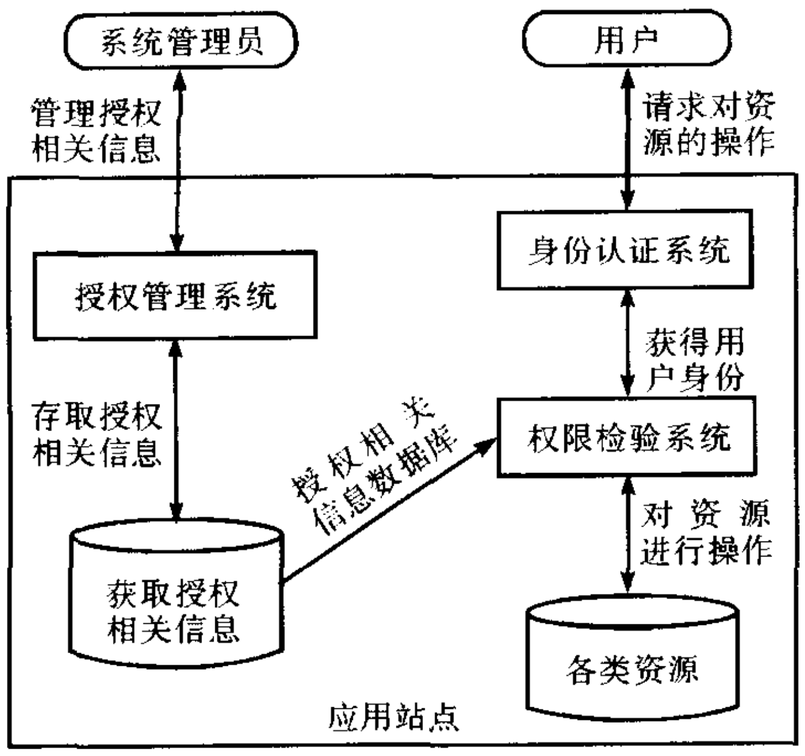

图2　系统概要

图 2 中的身份认证系统是“科学数据库及其应用系统”项目的另一子模块“单点登录认证系统”,单点登录认证系统完成用户认证的功能,只有被正确认证具有合法身份的用户才能被进行权限检验。

3.2.1 授权管理系统

在基于角色的访问控制框架中,系统管理员通过授权管理系统定义资源、操作和权限,并分配权限给角色,再给用户赋予角色。

授权管理系统主要包括以下七个子模块:(1) 资源定义模块;(2) 操作定义模块;(3) 权限定义模块;(4) 角色定义模块;(5) 角色权限管理模块;(6) 用户信息管理模块;(7) 用户角色管理模块

3.2.2 授权相关信息数据库

通过授权管理系统做授权管理而生成的与授权相关的所有信息全部存储在授权相关信息数据库中,以关系数据表的形式存在。

授权相关信息数据库主要包括以下七个数据:(1) AUTH _ RESOURCE;(2) AUTH _ OPERATION;(3) AUTH _ PERMISSION;(4) AUTH _ ROLE;(5) AUTH_ROLE_PERMISSION;(6) AUTH_USER;(7) AUTH_USER_ROLE。

3.2.3 权限检验系统

权限检验系统通过 web 容器的 Filter 组件实现并部署在应用站点上,它负责完成对用户权限的检验工作。权限的检验使用三元组判定法,一个用户 user 经过认证确定合法身份后,发请求希望对资源 resource 进行 operation 操作,请求可以被归结为一个三元组(user, resource, operation),权限检验的过程是检验该三元组的合法性,即通过查询授权相关信息数据库判断 user 对于 resource 是否有 operation 的权利,根据返回的结果(true 或 false)决定是否满足该用户的请求。

3.2.4 主要代码类

User:用户类,关联一个或多个角色。

Role:角色类,关联一个或多个权限。

Permission:权限类,关联(资源,操作)对。

Resource:资源类。

Operation:操作类。

在代码实现中采用开源的 *Hibernate* 工具进行对象和数据库存储之间的 OR 映射,对象通过 OR 映射保证数据持久性。代码不关心与数据库操作有关的底层具体实现,这些工作均由 Hibernate 来完成。

4 结束语

基于角色的访问控制(RBAC)是一种符合实际的访问控制模型,通过在用户层和权限层之间加入角色层,使系统具有了更灵活的安全策略。本文介绍的基于角色的访问控制框架是扩展的 RBAC 模型架构和系统实现。该访问控制框架仍然有许多有待改进的地方。比如资源划分的粒度可以更细;角色的转授权方面还有很大的空间有待研究。基于角色的访问控制框架将来的发展方向应该是做成通用性的访问控制框架,不仅仅应用在中科院“科学数据库及其应用系统”的项目中,而是更应该发展成为通过向外提供一些接口和扩展,可以应用到其它的很多领域和系统中。使其它的领域的系统能直接应用此框架就可以完成授权管理的工作,而不用再另做一套自己的授权管理系统。对于这种通用性的发展,还有待在今后做出更深入的分析和研究。

参考文献

[1] David F Ferraiolo, Ravi Sanehu. Proposed NIST Standard:

(下转第 107 页)

信源(300Mb/s)的位流数据,并对其进行位流业务处理，将其格式化为符合 AOS 协议标准的数据包(VCDU)，经合路器进行传送；将分路器送过来的 VCDU 进行解包后传送给用户。

在合路器端,高速位流数据首先被缓存在对应的 FIFO 中,导头生成单元产生版本号、航天器标识符、虚拟信道标识符、VCDU 计数器等主导头信息,链路控制单元根据 FIFO 中数据的存贮情况，通过选通多路选择器实现 VCDU 主导头和有效数据的拼接,从而输出完整的 VCDU。在分路器端,高速链路控制器从接收到的 VCDU 中提取出相关信息和原始数据,并将原始数据可靠传送给用户。

与其它低速链路控制器相比,高速链路控制器的数据率非常高,这给设计、调试和演示验证带来了困难。本方案采用高速、大容量的 FPGA 芯片,在满足足够大 FIFO 的情况下，还能取得足够高的数据处理速度。高速链路控制器和信道合路器/分路器做在同一个结构体中,以缩短器件间的连线。高速链路控制器与用户间,及高速链路控制器与合路器/分路器间采用并行接口,以确保数据稳定可靠的在用户与高速链路控制器间及链路控制器与合路器/分路器之间传输，同时避免 FIFO 之间数据的反复串并转换,提高了数据传输的速率。

6　测试及验证

高速数传系统可以支持 8 路链路控制器,其中 1 路支持 300Mb/s 的高速率，其它 7 路支持小于 50Mb/s 的任意速率。在合路速率为 500Mb/s 的情况下进行了验证。由于合路器在合路的过程中,需要在链路控制器传过来的 VCDU(虚拟信道数据单元)前增加一些同步信号以形成实际物理信道传送的 CADU(信道存取数据单元),增加了一些额外的开销,因此 8 路链路控制器输入的平均速率之和应略小于合路速率 500Mb/s，以保证传输中无数据帧丢失。测试结果表明,高速数传系统可以较好的保持合路与分路能力;各虚拟信道数据保持流畅,无丢失现象;每路数据的延迟时间都符合设计要求的最小时延;系统在发生意外事故的情况下,能够自动恢复正常。

7　结束语

本文介绍了基于 FPGA 的高速数传系统,该系统较好地模拟了 AOS 的数据传输体制,并且对数据传输中的一些关键技术进行了验证,为今后进一步的研究打下了基础。

参考文献

[1] CCSDS. Advanced Orbiting Systems, Networks and Data Links: Architectural Specification. CCSDS 701.0 -B -3,2001.6.

[2] 顾莹琦,谭维炽. 分包系统的星上数据源及其模型[J]. 遥测遥控,2001, 22(2): 26~31.

[3] 郑波, 张伟, 陈泓. 空间数传链路中虚拟信道动态管理的研究[J]. 空间电子技术,1999, (3):14~20.

[4] 谭维炽. 我国实现高级在轨系统的第一步[J]. 遥测遥控,1998,19(2):1~5.

[5] 孙白波.加强对 CCSDS 标准的认识与研究[J].遥测遥控,1999,20(4):52~57.

[6] National Semiconductor, "LVDS Owner's Manual", Spring 2000.

毛继志　男,(1976-),博士研究生。研究方向为数字通信与信号处理技术。

(上接第 103 页)

Role-Based Access Control. ACM Transactions on Information and System Security, August 2001, 4(3): 224~274.

[2] David F Ferraiolo, John F Barjkety, D Richard Kuhn. A Role-Based Access Control Model and Reference Implementation Within a Corporate Intranet. ACM Transactions on Information and System Security, February 1999, 2(1): 34~64.

[3] Sofia Tzelepi and George Pangalos. A Flexible Role-Based Access Control Model for Multimedia Medical Image Database System. ISC 2001,LNCS 2200, 2001: 335~346.

[4] Dianlong Zhang, Harald Lukhaub, Werner Zorn. A Role-Based Access Control Model and Implementation for Data-Centric Enterprise Applications. ICICS 2001, LNCS 2229, 2001: 316~327.

季永志　男,(1980-),硕士研究生。研究方向为大规模科学数据共享技术。

阎保平　女,(1950-),博士,研究员,博士生导师。研究方向为大规模科学数据共享技术、数据网格技术与应用、下一代互联网技术。

续　岩　男,(1981-),硕士研究生。研究方向为大规模科学数据共享技术。

吴开超　男,(1971-),高级工程师。研究方向为大规模科学数据共享技术。

沈志宏　男,(1977-),工程师。研究方向为大规模科学数据共享技术。