LIKE A PEARL IN THE SEA

THE PROCEEDINGS OF THE COMPUTER SCHOOL IN HUAZHONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

计算机海洋一粟

华中科技大学计算机科学与技术学院论文集(第三集)

计算机科学与技术学院　编

计算机海洋一粟

华中科技大学计算机科学
与技术学院论文集（第三集）

计算机科学与技术学院 编

· 内部交流 ·

序

以计算机科学与技术为中心的信息技术革命引发了社会的大变革，正使人类社会发展从工业社会阶段走向信息社会阶段，工业化进程即将完成 ，信息化进程已经拉开序幕，这是一个不以人的意志为转移的大趋势。信息革命、信息化将是这个时代的重要科技主旋律和主题词。

近些年来，我院教师承担了一批国家计划研究项目，包括教育部中国教育科研网格计划、国家基础研究重大项目(973 计划)两个项目的牵头单位，高技术项目(863)、国家支撑计划、国家自然科学基金项目、部和省重点项目及国际、国内合作项目，取得了一批重要科研成果，有的成果获得国家自然科学奖、国家技术发明奖和国家科技进步奖。

本书收录了我院2004年在相关重要刊物和国际学术会议上发表、被 SCI（Science　Citation Index）收录的论文。这些论文涉及信息与网络存储、集群与网格计算、流媒体、图形图像处理、实时系统、软件工程、智能代理、算法理论及优化、生物信息学等方面所做的工作。这些论文对计算机科学与技术在若干方向上的理论研究与应用实践有其较好的指导意义和参考价值。本书的出版，有利于促进我们与国内外同行的学术交流和合作研究，共同为推动计算机学科的发展做出新的贡献。

华中科技大学教授
计算机科学与技术学院院长 金海

二00七年十月

前　言

本书收录了我院金海、冯丹、谢长生等10几位老师、博士的学术论文三十五余篇以《计算机海洋一粟-----华中科技大学计算机科学与技术学院论文选集第三卷》冠名成书出版(2004年出版第一卷、2005年出版第二卷)。本卷论文都是2004年被SCI（Science Cited Index）收录的，较集中地反映了这段时间我院在计算机系统结构、计算机软件与理论、集群与网格、以及计算机应用方面所取得的一些研究成果。一方面是我院研究工作一个侧面的小结与回顾，另一方面是以期更有利于与海内外同仁们进行交流与切磋，共同为计算机科学与技术事业的发展作出贡献。

本书的编辑，得到作者们的鼎力相助，王炎坤教授为本书的出版精心策划，李久红同志为本书的组稿、论文收集和编辑做了大量工作，也得到我院全体师生的支持，在此一并表示感谢。

本书的出版，得到华中科技大学出版社印刷厂支持，在此表示感谢。

二〇〇七年十月

A RBAC Based Grid Access Control Architecture*

Weizhong Qiang, Hai Jin, Xuanhua Shi, Deqing Zou, and Hao Zhang

Cluster and Grid Computing Lab Huazhong University of Science and Technology, Wuhan, 430074, China {wzqiang, hjin}@hust.edu.cn

Abstract. Because the distribution of services and resources in wide-area networks are heterogeneous, dynamic, and multi-domain, security is a critical concern in grid computing. This paper proposes a general authorization and access control architecture, RB-GACA, for grid computing. It is based on classical access control mechanism in distributed applications, Role Based Access Control (RBAC). We also use a kind of standard policy language as the presentation of access control policies to provide a general and standard support for different services and resources.

1 Introduction

Due to the characteristics of heterogeneity, dynamic and organization self-governed, there are many challenges to be solved for grid technology, such as authentication, authorization, and resource discovery [1]. Authentication, authorization and audit (AAA) [2] are general issues for

* This paper is supported by National Science Foundation under grant 60125208 and 60273076. W. Qiang et al.

grid environment. Most of the research interest presently is on authentication, and little is on authorization and access control.

Authentication is the first problem. The Globus project proposed and developed the Grid Security Infrastructure (GSI) [3] that is authentication architecture for grid computing. This mechanism provides single-sign-on approach, cross-domain protocol and some convenient security API for grid applications.

Authorization is a challenge mission. The authorization mechanism of Globus is very inflexible and coarse-grain, which uses a mapping table in every grid node that only maps the global name (a ticket or certificate) [4] into a local name (login name or user ID). A flexible, fine-grained, and general authorization and access system is significant for grid computing.

In this paper, we present a role based grid access control architecture (RB-GACA). We provide a flexible framework for policy management that treats the whole grid as a series of independent, dynamic domains. The policy evaluating and decision making is based on role-based access control (RBAC). XACML [5] is used as the presentation policy language.

The paper is organized as follows. First, we present some related works about RBAC and some classical access control approaches. Then we describe our access control model. Our system architecture and access control policy expression are presented later. Then we give the performance analysis. Finally, we conclude our paper and present some future considerations.

2 Related Works

Role Based Access Control model (RBAC) [6][7] is an alternative approach to traditional access control model discretionary access control (DAC) and mandatory access control (MAC). In MAC and DAC access control models, subjects and objects have direct relationships, which cause oppressive burden of security management.

In RBAC, there are three layers, users, roles and permissions. Users are assigned roles, and permissions are assigned to roles. Because users and access permissions are separated, the management of access control is more expediently and costs less [8]. In RBAC, the user-role relationship is more dynamic than the role-permission relationship.

Permis [9] is a policy driven RBAC Privilege Management Infrastructure (PMI). The policy is written in XML and stored in X.509 attribute certificates (AC) in the local LDAP directory. The credentials may be widely distributed. The core model of Permis is Access Control Decision Function (ADF).

Akenti [10] is an access control architecture that addresses issues that all the resources are controlled by multiple authorities. The Akenti policy engine gathers user-conditions certificate and attribute certificates, and grants access to a resource by verifying these two types of certificates. In Community Authorization Service (CAS) [11], the owners of resources grant access to a community account as a whole. The CAS server is responsible for managing the policies that govern access to a community's resources. It maintains fine-grained access control information and grants restricted GSI proxy certificates to the users of community. K. Keahey et al [12] propose a fine grain authorization system in grid by modifying GRAM of Globus. But it has the limitation of management and scalability. L. Ramakrishnan et al [13] also present an authorization infrastructure by providing authorization at the component interface. But their special objects are component-based grid applications.

3 Model of RB-GACA

The access control model of our system is based on the RBAC model presented in NIST [5]. As the NIST model is a complete proposal and is too complex for our system, we make some simplification to adapt it to our system. In RB-GACA access control model, all the access control domain of the grid is naturally divided into a series of domains, each of which is an autonomous region.

RB-GACA access control model consists of following components:

1) *U, R,OP,OB* are sets of users, roles, options and objects in an access control domain, respectively

2) $P = 2^{(OP \times OB)}$ is a set of permission in the domain

3) $PA \subseteq P \times R$ is a relationship set that defines permission to role assignment in the domain, a role in R can have the operating permission defined in PA

4) $UA \subseteq U \times R$ is a relationship set that defines role to user assignment, a user in U can enable the roles defined in UA

We also used some administrative operation definitions in RB-GACA, which includes add/delete user, add/delete role, assign/de-assign user-to-role assignment, and grant/revoke permission-to-role assignment. Definition 1 is one of the definitions.

Definition 1. A domain manager may execute the AddUser(user : NAME) operation if it is a new user : NAME ($user \not\subset U$) to the domain and the user is authenticated by the domain, authenticated(user) = true . The result is that the user : NAME is added to the set U , $U'=U \cup \{user\}$.

4 System Architecture

For an authorization and access control infrastructure, two factors must be predefined: the repositories location of the access policies, the relative location of the policy decision point (PDP) and the policy enforcement point (PEP). For the first factor, there are two choices: the policies may be stored in one repository centrally or stored in several repositories distributed. To the first choice, the management is convenient but it lacks scalability and reliability. The later choice has the contrary characteristics. As to the second factor, arranging PDPs locally with the distributed PEPs have the benefit of scalability and reliability, but it is harder to manage. However, arranging PDP centrally has the benefit of management and the limitation of scalability.

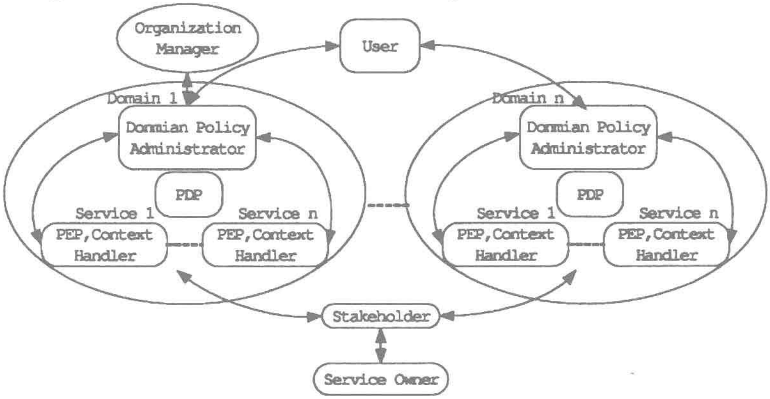

Fig. 1. RB-GACA System Architecture

Figure 1 shows the overall RB-GACA system architecture. The grid is divided into a series of logical domains each of which is an independent access control region. Our system architecture predefines the two factors mentioned above. Repositories location of the access policies is in every domain separately. PDP is domain based, and PEP is service or resource based. If a manager of an organization wants to put some access control polices on the resources, he may initially construct an access control domain to apply the RB-GACA system to these services and resources.

This arrangement is a tradeoff between central and distributed approaches. The advantages are convenient and scalable. If a domain is too huge to be managed, the domain may be divided into two or more domains by the domain policy administrator.

4.1 System Components

As shown in Figure 2, RB-GACA access control architecture is composed of four components: Policy management modules, policy decision modules, policy enforcement modules, and context handler module. For policy management module, there are two sub-modules according to the functions requirement from core RBAC in NIS, Element Sets management agent and Relationship maintenance agent.

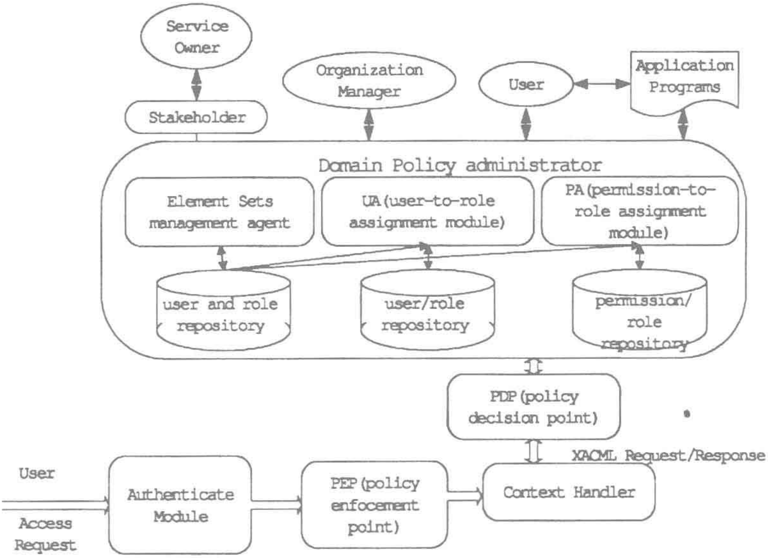
Service Owner
Organization Manager
User
Application Programs
Stakeholder
Domain Policy administrator
Element Sets management agent
UA(user-to-role assignment module)
PA(permission-to-role assignment module)
user and role repository
user/role repository
permission/ role repository
PDP(policy decision point)
XACML Request/Response
User
Access Request
Authenticate Module
PEP(policy enfocement point)
Context Handler

Fig. 2. Access Control System Components

Element Sets management agent is a domain based module for element management. In NIST definition, users, roles, operations, and objects are basic element. These elements may be divided into two classes: operations and objects, users and roles. Operations and objects are relatively static which may be predefined when RBAC system is deployed to a resource or service. For example, in a virtual instruments grid system, the objects may be predefined as some instruments, such as a telescope. The operations may be predefined as steer object, view object, and some other basic operations. However, users set and roles set are relatively dynamic which may be managed by the domain policy administrator.

From Fig.1, the functions of creation and deletion of roles or users are provided by AddUser and DeleteUser operations for users and AddRole and DeleteRole operations for roles. The elements set is stored in the user and role repository.

Relationship maintenance agent module accomplishes two policy assignment operations: user-to-role assignment (UA) and permission-to-role assignment (PA). For scalable and dynamic benefit, it is also domain-based. It includes AssignUser and DeassignUser operations for UA and GrantPermission and RevokePermission operations for PA.

PEP and authenticate module are both service based, which means that we must separately enforce each type of service. In our preliminary research, we have modified the Globus gatekeeper authorization call-out interface to add the policy enhancement point for job submission service, the GridFtp module for ftp service and Information search module for information search service.

Context Handler is an entity that converts access requests in native request format to XACML canonical form and converts authorization decision in XACML canonical form back to native response format. In our prototype implementation, we convert the native request expression such as Globus RSL request into XACML presentation.

PDP gets the XACML request from Context Handlers and responses according to the set of polices from domain policy administrator. Polices are embedded in X.509 attribute certificates [14] which assure secure binding of polices with the domain policy administrator. Sun XACML implementation [15] is used in our initial implementation.

4.2 Scenarios of RB-GACA

4.2.1 Scenarios of Management Roles

There are two different types of management roles, Organization Manager and Service Owner. Organization Manager is domain based which is responsible for policy administration of the whole domain. Service Owner is service based which is responsible for services or resources. If an organization or a research group wants to grant access control mechanism to the services or resources, the Organization Manager may construct an authorization and access control domain.

Organization Manager must predefine some basic users set, roles set, operations, and the objects through the domain policy administrator, which is a GUI daemon process. The domain policy administrator daemon utilizes Element Set management agent to store these users and roles elements into the user and role repository.

Organization Manager must also predefine some basic policies through the domain policy administrator, which utilizes Relationship Maintenance agent to store the UA policies into user/role repository and the PA policies into permission/role repository.

The Service Owner uses the stakeholder, which is also a GUI daemon process, to interface with the domain policy administrator and get the view of the user set and role set from the user and role repository. The Service Owner references the above view from stakeholder and makes decision of the access control policies for the objects it owns, and stores polices into the user/role repository and permission/role repository through the stakeholder. A Service Owner has the permission to add, revoke or modify his own policies. He also has the permission to consult with the domain policy administrator, which notifies the Organization Manager to add some new roles or users into the user and role repository.