一种基于域的递增式策略部署模型

何再朗[1]，田敬东[1]，张毓森[2]

(1. 解放军理工大学通信工程学院 南京 210007; 2. 解放军理工大学指挥自动化学院 南京 210007)

【摘要】在介绍策略、域和策略目标等基本概念的基础上，提出了一个通用性的策略部署模型，该模型采用基于域的分层结构，把策略存储在离策略目标最近的域中，并以递增式算法进行收敛变化。模型独立于具体的策略底层实现机制，并可在混合策略环境中使用，具有良好的多适性。

关 键 词 策略; 域; 递增式; 部署模型
中图分类号 TP393 文献标识码 A

An Incremental Policy Deployment Model Based on Domain

HE Zai-lang[1]，TIAN Jing-dong[1]，ZHANG Yu-sen[2]

(1. Institute of Communication Engineering, PLA University of Science & Technology Nanjing 210007;
2. Institute of Command Automation, PLA University of Science & Technology Nanjing 210007)

Abstract Policy based management is an effective approach to manage distributed system and large scale system. With the basic concepts of policy, domain and policy target introduced, a general policy deployment model is purposed which based on domain layer architecture. It assumes that policies are stored in the most close domain to target, and the whole system changes according to the increment arithmetic. The general model is independent of the underlying concrete policy enforcement mechanisms and can be employed in mixed policy environments.

Key words policy; domain; incremental; deployment model

所谓“策略”，就是一组相对持久的、说明性的规则集合，这些规则用于指导和决定如何管理、分配和控制系统资源，并约束系统做出决策的过程。较之传统的管理方法，策略管理有以下优势：(1) 较强的伸缩性。管理员可根据系统的整体状态统一制定管理方案，以便最大限度地保持系统的一致性，适合对异构系统进行管理；(2) 较好的灵活性。当系统状态发生变化时，管理员通过修改策略，即可在不中断系统运行的前提下实现系统功能的重构，修改策略甚至可在高层策略的约束下自动完成；(3) 较高的效率。策略屏蔽了具体的物理设备和技术细节，管理员负担得到减轻，效率得到提高。

近年来，策略管理广泛应用于分布式系统的网络安全、服务质量控制等多个领域，取得了一定的成效。然而，目前有关策略的研究工作过于集中在策略规范、信息模型和策略实施等方面，而忽略了部署机制的研究，已有的机制效能低下，成为管理系统的瓶颈，严重制约着策略的应用。鉴于此，本文提出了一个基于域的递增式策略部署模型(以下简称模型Ⅰ)。

1 基本定义

借鉴文献[1-2]中的工作，可将策略做如下定义：

定义 1 策略是一组描述任务和规则的集合。可形式化表示为：*policy*::={*policy-id*}<*type*>{*trigger*}<*subject*><*object*><*action-list*>{*constraint*}{*exception*}。其中，*policy-id*：策略标识符，在系统中具有唯一性；*type*：策略类型，包括授权(*auth*)、委托(*deleg*)、职责(*oblig*)和节制(*refrain*)等；*trigger*：主动策略被激活的条件，采用一组事件描述符来描述；*subject*：策略主体集合，是策略的执行者；*object*：策略客体集合，是策略施加的对象；*action-list*：策略动作集，定义了策略的动作行为；*constraint*：策略约束条件，只有当约束条件满足时策略才会被执行；*exception*：策略异常处理，定义了策略未能顺利实施时的补救措施。

收稿日期：2004 – 06 – 18
基金项目：国家863高科技发展计划基金资助项目(2002AA141090)
作者简介：何再朗(1976 –)，男，博士生，讲师，主要从事策略管理方面的研究.

定义 2 域是具有某些相同属性的对象的集合，是数据集中管理的单位。域的成员有两种：对象(*object*)和子域(*sub-domain*)。可表示为：*domain* ::= [*object* {, *object* |, *sub-domain*}]。

如*A*域为*B*域的子域，简记为$A < B$。系统中最大的域称为根域，记为*root*。系统中所有的域构成的集合记为*D*。需要指出，域不是简单的树结构，而是一种允许交叉的树，但是不允许出现环。

定义 3 策略目标：根据策略的类型，主动策略(职责、节制)的目标是其主体对象集合，被动策略(授权、委托)的目标是其客体对象集合。

$$target(policy)=\begin{cases} policy.subject & if\ \ policy.type = oblig、refrain \\ policy.object & if\ \ policy.type = auth、deleg \end{cases}$$

定义 4 策略目标域：包含策略目标的最小的域。即策略目标域*target-domain*满足：$target(policy) \subseteq target\text{-}domain$，且$\neg\exists\, domain \in D$，使得$target(policy) \subseteq domain < target\text{-}domain$。

由于域之间可能存在交叉关系，所以一个策略的策略目标域有可能不唯一。

定义 5 策略存储域：存储策略原始记录的唯一的域。

规则 过滤策略：将策略目标与域的交集作为新的策略目标而构造出的策略。

$filter(policy, domain) \rightarrow policy1$使得$target(policy1) = target(policy) \cap domain$

2 基于域的递增式策略部署模型

2.1 模型的系统结构

受文献[3-5]的启发，本文提出如图1所示的策略部署系统结构。图中整个系统被划分为多个域进行管理，每个域由4个基本功能模块组成，策略管理器为可选模块：(1) 策略决策点(Policy Decision Point，PDP)：域内的决策中心，负责访问策略库中的策略，并根据策略信息做出决策，然后将策略分配至相应的策略执行点。同时，它还负责策略的检查和确认，以保证策略之间不会发生冲突。(2) 策略执行点(Policy Enforcement Point，PEP)：接受策略管理的实体，负责执行由PDP分配来的策略，可加载不同的策略实施机制。另外，它还负责向PDP发送信息，通知PDP系统的变化以及策略的执行情况。根据策略类型的不同，主动策略的PEP为策略管理代理(Policy Manage Agent，PMA)，被动策略的PEP为访问控制器(Access Controller，AC)，这正是定义3中策略目标根据策略类型而定的根本原因。(3) 策略库：用于存储策略信息，通常为目录服务器或数据库服务器。(4) 策略管理器：实现策略编辑和监控功能的模块，管理员通过其提供的人机界面来维护策略。PDP和策略库之间则通过LDAP(Lightweight Directory Access Protocol)协议进行存取，其他模块包括PDP之间则通过COPS(Common Open Policy Service)协议联系。

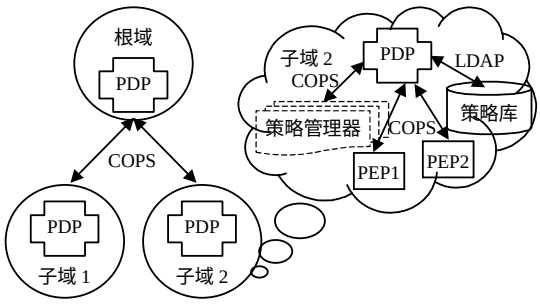

图1 基于域的策略部署模型的系统结构

模型Ⅰ有如下假定：(1) 策略存储域必须同时是策略目标域；(2) 存储域的父域存储策略的引用拷贝；(3) 存储域的子域存储策略的引用拷贝和其对子域的过滤策略；(4) 所有策略目标域中标志符最小。由定义4可知，假定(1)保证了策略的原始记录存储于离策略目标最近的域中；假定(2)和(3)则保证了任意域均保存有应用在该域对象上的全部策略。这种部署模式使得PDP在查找和分析检查策略时可以直接在本域进行，不必再到其他域去搜寻策略，花费代价大幅减小。

2.2 递增式策略部署算法

系统在运行的过程中可能会发生策略的变化和系统结构的变化。此时，用户通常期望系统在即时适应变化的同时保证运行稳定而不被中断。递增式的策略部署算法就是根据这个目标而提出的。所谓“递增式”，实质上是指系统的任何变化都有一套相应的算法保证系统能快速地收敛到稳定状态。

2.2.1　策略变化的递增算法

策略变化时，其部署过程如图2所示。算法描述如下：(1) 定位存储域。管理员在某个域的策略管理器上提出操作策略的请求，系统计算出策略的存储域，并检查操作域是否是存储域的被信任域，即操作域是否有操作存储域的权利。域的信任关系可由父域和子域的关系隐式说明，也可由存储域的某个策略显式指明。(2) 分配标志符。当添加新策略时，由存储域负责分配一个全局唯一的策略标志符。(3) 检查策略。对于添加和修改策略，需要将新策略与存储域中原有策略进行比对检查，以保证策略的正确性。检查包括语法和语义两个方面，语义检查又包含上下限检查、关系检查、一致性检查、支配性检查和可行性检查等等。(4) 存储策略。检查策略后，存储域负责存储该策略。(5) 部署策略。存储域将该策略的引用拷贝发送给所有的父域(父域可能不止一个)，每个父域也执行同样的操作，直至根域；存储域将该策略的引用拷贝和对子域的过滤策略发送给所有的子域，每个子域也执行同样的操作，直到非域对象。

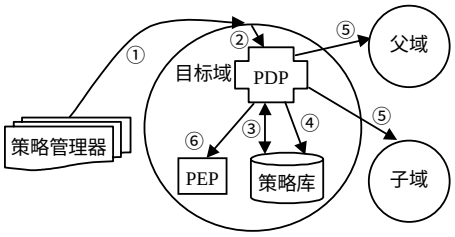

图2　策略变化的部署过程

2.2.2　系统结构变化的递增算法

系统结构变化包括域的添加、删除和修改，下面给出添加域的算法，删除和修改与之类似，不赘述。1) 检查域结构。添加域之后，检查新的系统域结构，保证没有环的出现。2) 部署来自父域的策略。(1) 拷贝所有父域中的所有原始策略和过滤策略，并对新域进行过滤。若过滤前后的策略一致，且父域中的记录为原始记录，则通知父域将该策略的存储位置迁移到新域。若不一致，简单地存储引用拷贝和过滤后策略。(2) 将父域策略的引用拷贝和过滤策略发送给所有的子域。每个子域也执行同样的操作，直到非域对象。(3) 部署来自子域的策略。将子域中所有的策略拷贝过来，简单地存储引用拷贝；将子域策略的引用拷贝发送给所有的父域。每个父域也执行同样的操作，直到根域。

3　域的重叠问题及解决方案

域的重叠问题指的是某个策略有两个或两个以上的策略目标域[6]，这些策略目标域都是策略存储域的候选，此时需要根据某种规则来做出选择，选用其中的一个域作为策略存储域来存储策略的原始记录，而另外的域则存储策略的拷贝和引址。域的重叠问题可用如下计算策略存储域的算法来解决：(1) 在每创建一个新域的时候，给它分配一个唯一的标志符。(2) 从根域开始，判断哪个子域包含策略目标。向包含的子域发送计算存储域的请求，最后从所有返回的结果中选择标志符最小的一个作为结果；若所有的子域均不包含目标，则返回当前域。(3) 收到计算请求的子域执行和父域同样的操作。

显然，该算法实际上是一个分布式的递归算法，其正确性来源于策略存储域符合假定(4)。但是，假定(4)又引入了一个新的问题，即添加域可能导致策略的迁移问题。如图3所示。从图中可看出，新域和兄弟域的成员完全一致，按照假定(1)，新域作为兄弟域中策略的存储域是完全合格的，如果此时新域的标志符又小于兄弟域，按照假定(4)就会有策略迁移情况的发生。因此，需要对2.2.2中的算法加以修正。修正后的算法如下：(1)～(3) 同前。(4) 新域向所有的子域发送请求，要求传送子域的父域(即兄弟域)的标志符。(5) 新域向标志符大于自己的兄弟域发送策略迁移的请求。(6) 兄弟域判断存储的原始策略是否满足迁移条件，如果满足，将原始策略发送到新域，并向父域和子域发送策略迁移改变的消息。

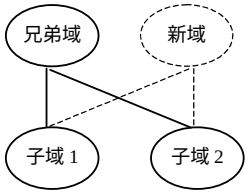

图3　兄弟域中存储的策略有可能要迁移到新域

4　模型分析及仿真实现

与面向对象的策略部署模型(以下简称模型 II)的集中策略部署方式不同[3]，模型 I 将系统的功能模块建立在域的基础上，以域为基本单位来实现策略的部署，策略被分散部署在离目标最近的域中。

域的引入使得模型 I 在应用于大规模分布式系统时具有以下优势：(1) 域策略适用于域内所有对象和子

域，当对象转移到新域时会自动受限于新域的策略，无须手工修改策略；(2) 根据需求的不同，域有多种灵活的分割方式，如按地理位置，对象类型、职能和权利，或是管理员的特殊需求等等；(3) 域的分布式和多备份特性同时还提高了管理系统的可靠性。这些优势为管理员对大系统内数以万计的对象制定管理策略和有效管理提供了极大的便利。

模型Ⅰ在部署效能上也较模型Ⅱ有很大的提高。本文利用GPSS语言分别实现了模型Ⅰ和模型Ⅱ的原型，并以文献[5]中图4.2介绍的包含13个节点的域结构为场景做了仿真实验。假定对策略库的一次访问要花费一个单位的时间，在系统策略总数分别为100、200和500时，分别测试模型Ⅰ和模型Ⅱ的策略部署(不含检查)时间和策略检查时间，仿真结果如表1和表2所示：

表1 平均策略部署(不含检查)时间

策略总数	模型Ⅰ	模型Ⅱ
100	3.923	1
200	3.912	1
500	3.915	1

表2 平均策略检查时间

策略总数	模型Ⅰ	模型Ⅱ
100	7.691	99.365
200	15.385	200.852
500	38.458	499.631

表1中说明，策略部署(不含检查)时间与系统的策略数目是无关的，进一步研究发现模型Ⅱ是固定值，模型Ⅰ则与系统的域结构相关；从表2中看出，策略检查时间与策略数目成正比。虽然模型Ⅰ在策略部署时间上比模型Ⅱ增加了近3个单位的时间，但是在策略检查时间上获得了相应的收益，并且策略数目越多，收益越大。如在仿真时计入系统运行时策略的动态检查，其实际收益值应该更大。

此外，模型Ⅰ仅涉及策略的类型、主体和客体三个基本属性，这使得它与具体的策略语言呈现出弱耦合性。PEP模块封装了策略的底层实施机制，如果在某个PEP上载入多种策略实施机制，或在不同的PEP上载入不同的策略实施机制，模型即可应用于混合策略环境中，具有良好的多适性。

5 结 束 语

在大型分布式系统管理中，如何正确而有效地部署策略是一项复杂的工作。分布式系统的复杂性通常需要进行分层式的管理，而域作为一种灵活使用的组织资源的手段，恰恰适应了这种需求。因此，本文提出了基于域的递增式策略部署模型，其基本思想是在具有域层次的分布式系统中将策略部署于离策略目标最近的域中，并分发到所有相关的父域和子域，在需要的时候由各个PDP将策略分配到相应的PEP上。本文算法虽然是按序给出，但在实现时均可采用分布式的实现机制。

在策略部署过程中，可能会发生策略的语义冲突或者存储共享冲突，由于策略的多备份特性可能导致策略的不一致，这都是下一步的研究方向。

参 考 文 献

[1] Rene W. Using a classification of management policies for policy specification and policy transformation[C]//In: A Sethi, Y Raynaund, F Faure-Vincent. Integrated Network Management IV. London: Chapman Hill, 1995:44-56.
[2] 李 莉, 任秀丽,栾贵兴. 基于策略的分布式网络管理系统[J].东北大学学报, 2002, 23(6): 515-518.
[3] 杨海松. 基于策略管理的分布式、动态网络安全模型:[D].合肥:中国科学技术大学, 2003:86-99.
[4] Dulay N, Lupu E, Sloman M, et al. A policy deployment model for the ponder language[C]// In: IEEE/IFIP International Symposium on Integrated Network Management. Seattle: 2001:529-544.
[5] RFC 3060. Policy core information model - version 1 specification[S]. 2001.
[6] 吴礼发. 基于SNMP的网络管理系统分布式策略的研究[J].电讯技术, 2001, 41(3):95-98.

编 辑 刘文珍