网格计算中面向虚拟组织的高可用安全目录服务研究*)

赵曦滨[1] 吴 雷[1] 裴 军[2]

(清华大学 软件学院 北京 100084)[1] (江苏长江电器股份有限公司 江苏 镇江 212211)[2]

摘 要 本文针对虚拟组织对目录服务的要求,结合检查点归档算法、SOAP协议及多机部署方案,设计并实现了基于Web Services的轻量目录服务,并针对目录服务自身的安全做了基于代理服务器和服务迁移机制的安全扩展。这种目录服务可以在保持服务安全性和可靠性的同时为虚拟组织提供良好的可用性。

关键词 网格计算,分布式系统安全,访问控制,授权,目录服务

Research on the High Availability of Secure Directory Service for Virtual Organization in Grid Computing System

ZHAO Xi-Bin[1] WU Lei[1] PEI Jun[2]

(School of Software, Tsinghua University, Beijing 100084)[1] (Jiangsu Changjiang Electric Co., Ltd. Zhenjiang, Jiangsu 212211)[2]

Abstract This article presents a lightweight directory service based on Web Services, with the combination of checkpointing algorithm, SOAP protocol, and multi-machine-based deployment. In addition, the service migration mechanism is also designed to protect directory service itself. This directory service has the attractive availability while keep the security and reliability.

Keywords Grid computing, Distributed system security, Access control, Authorization, Directory service

1 引言

作为一种新型分布式计算基础设施,网格计算系统近年来在科学计算和商业领域得到了广泛应用。网格计算关注广域地理分布环境下动态与大规模的资源共享,因而更适应于实现动态环境下的海量数据处理以及面向复杂任务的协同工作。

网格计算系统中的资源共享和协同工作建立在虚拟组织基础之上。虚拟组织实际上是对具有复杂内联关系的分布式系统的抽象描述,而网格计算的目标就是在动态、多成员的虚拟组织中实现资源共享与协同工作。在虚拟组织的生成与运行过程中,必须合理地选择参与者和资源,并以安全、灵活的方式将它们组织起来,以完成特定任务。随着网格计算的应用不断从科学计算转向电子商务应用,安全可控的资源共享及运行模式对虚拟组织来说越发重要,因此,基于虚拟组织的访问控制对网格计算系统意义重大。

目录服务的设计与实现是支持VO协同工作和资源共享的重要环节。在VO支撑平台系统架构中,目录服务起到了连接纽带的作用。不论是在VO形成时的选择与定位、注册时的系统策略实施,还是运行时的成员管理、访问控制,都需要目录服务的帮助。网格计算系统的拓扑结构和操作模式的特殊性要求目录服务以轻量级的方式建立在通用的分布式计算架构之上,能够具备良好的可靠性和可用性,并且可以充分利用网络带宽以支持虚拟组织的运作。本文针对虚拟组织对目录服务的要求,设计并实现了基于Web Services的轻量目录服务,并针对目录服务自身的安全做了基于代理服务器和服务迁移机制的安全扩展。

2 虚拟组织访问控制系统中的目录服务

2.1 虚拟组织访问控制系统对目录服务的要求

由于虚拟组织自身的特点,在实现访问控制时对目录服务机制有特殊的要求。

首先,对于VO管理平台来说,目录的准确性更加重要。如果存放在目录中的信息不准确,就不能提供受控的资源共享服务。因此,VO中的目录服务要求具备更高的可靠性,即使当面临服务器或网络瘫痪时,也应尽量保持服务的正确性和数据的准确性。

其次,各种访问控制相关服务在运行时的动态绑定对虚拟组织来说也是非常必要的。VO的运行主体、资源对象和控制策略都处在动态变化的过程,因此每当VO需要进行信息的查询、检索与定位时,目录服务都必须可用。在这里,目录服务的可用性包含两方面的含义:

·在任意需要访问的时刻,目录服务都应该是可以访问的。一方面,VO的成员在动态变化;另一方面,基于Internet的网格计算基础设施总会面临各种突发的机器及网络故障。所以,VO中的目录服务应当在系统重启或成员变化时快速恢复对客户端的响应能力。

·目录服务的理想应答时间应该尽可能少且平滑。一方面,应答时间的降低不仅能够提高VO的执行效率,而且有助于减少带宽拥塞;另一方面,在VO的访问控制实现过程中,大部分资源定位的路由算法往往关注系统的拓扑结构和连接协议,忽略了目录服务的访问能力和响应时间。但是,如果目录服务的响应时间波动太大,VO中对资源及服务的定位能力和定位效率都会降低。此外,在一些对服务的实时性和稳定性要求较高的军用和商务应用中,系统响应的振荡将对应用产生很大的影响,所以希望目录服务的响应时间应该尽可能平稳,避免出现振荡。

再者,作为服务请求和检索的纽带,目录服务必须和大量的客户端通信,而这些客户端有可能位于多种不同的系统环

*)国家"863"项目(No.2003AA148020,No.2003AA413031)。**赵曦滨** 博士后,主要研究方向为计算机系统安全、软件工程;**吴 雷** 硕士研究生,主要研究方向为目录服务、软件工程;**裴 军** 工程师,主要研究方向为计算机辅助设计,电子商务。

境中，因此目录服务的设计应当尽量采用被其他系统接受并使用的通用标准或协议。

另外，更加适应 VO 访问控制的目录服务必须是轻量级的。早期的典型协议标准是目录访问协议(Directory Access Protocol, DAP)[1]，它被用于对 X.500 目录的访问中。然而 X.500 技术有一个严重的缺点，即太复杂，对系统处理能力要求高。从 X.500 中分离出来的轻量目录访问协议(Lightweight Directory Service Access Protocol, LDAP)使目录对于广泛多样的服务器或应用程序来说更可用[2~4]。VO 实体的自治性要求目录服务不能集中管理，必须分布在系统的相关部分。在这种情况下，如果不采用轻量级的架构，目录服务的实现和维护所需要的开销是无法承受的。除此之外，随着移动计算设备的大量使用，很多电子商务模型都采用移动设备作为交易部件。因此，越来越多的网格计算系统被设计为允许移动用户和移动资源的参与[5]。可以想像，在这些移动实体上部署类似 Oracle 这样的大型数据库产品来提供目录服务是不现实的。可见，VO 的访问控制需要轻量级的目录服务。

最后，作为支持 VO 访问控制的关键模块，目录服务自身的安全也极为重要。在变换多端的网络环境下，资源管理的安全性成为 VO 资源管理急需解决的问题。例如，对目录服务自身来说，拒绝服务(Deny of Service, DoS)就是一种非常有威胁的攻击手段。这种攻击会导致资源的匮乏，无论计算机的处理速度多么快，内存容量多么大，互联网络的速度多么快都无法避免这种攻击带来的后果。

由此可以看出，为支持虚拟组织的访问控制，理想的目录服务应该基于像 Web Services 这样的全球化分布式基础架构，并且具有轻量级的体系结构。同时，可用性和可靠性是虚拟组织目录服务的两个重要运行指标。此外，目录服务自身的安全性也必须得到保障。

2.2 相关的研究工作

在 Globus 项目中，元计算目录服务(Metacomputing Directory Service, MDS)[6]是相当重要的组成部分。网格计算环境中存在各种资源，它们在地理上分散，又可以动态地加入或离开不同的虚拟组织，如何方便、高效地使用各种资源是 Globus 项目必须解决的问题。为此，在 Globus 项目中提出了 MDS，它主要是一种基于网格计算环境的信息服务框架，面向网格计算环境中数目巨大、地理上分布且具有动态性的各种资源和服务。MDS 的内容主要包括资源(服务)发现、资源(服务)描述和资源(服务)监视与更新等。在实现上，MDS 的核心部分是 OpenLDAP 服务器，并借助 LDAP 协议来实现网格信息访问与存储。

尽管 MDS 借助 LDAP 可以有效地把虚拟组织中的资源和服务组织起来，提供网格计算应用所需要的信息服务，但它借助于公开的目录服务软件 OpenLDAP，主要关注于如何表达、组织和访问目录中的相关信息。然而，不论目录协议以何种类型被使用，目录信息本身应该以某种方式存储和管理，而目录服务自身信息的存储与管理方式将直接影响到目录服务是否能向 VO 提供上节所描述的那些性能指标。

为了在目录服务的实施成本、可用性和可靠性之间找到一个平衡点，Birrell 提出了一种实现高可靠性和可用性的轻量目录服务技术[7]。在这种技术中，目录数据保存在虚拟内存中，而在硬盘上维护数据的映像(Image)，即一个检查点(Checkpoint)文件和一个日志(Log)文件。Checkpoint 文件用于维护整个数据在某一时刻的状态，而 Log 文件则记录了这个时刻之后的所有对内存中数据的操作。当系统重新启动的时候，可以通过把当前 Checkpoint 文件中记录的信息恢复到内存中，然后再把日志文件中记录的操作，重新应用于内存中的数据，从而得到最新的目录数据。该处理又称为归档过程。Birrell 方法在一定程度上提高了目录服务的可靠性，但是，基于该方案的目录服务无法保证其可用性。有两方面的原因限制了它的可用性，即生成检查点的时间要求和系统失败后用于重启的时间要求。新检查点的生成不仅非常消耗时间，还要求暂停更新操作，会对目录服务的可用性产生影响。但是，如果采用减少归档次数的办法来提高可用性，又会使日志变得很大，一旦由于某种原因导致系统重启，重启时间会加长，也会影响系统的可用性。可见，Birrell 方法要求系统管理员在系统重启时间和归档频率之间做出权衡。虽然这种设计简单、高效、可靠，但是当更新操作相对频繁时就不能有效地保障服务的可用性。

为了提高目录服务的可用性，Lam 等人基于 Birrell 的思想提出了一个重要的改进方法[8]。该目录服务的设计思想就是在生成 Checkpoint 时并不需要停止服务，而是由目录服务器产生一个新的进程，在后台生成新的 Checkpoint 文件，同时产生一个新的 Log 文件继续记录来自客户端的操作。这种方法的优势在于：生成新的 Checkpoint 时是在一个新的地址空间里面进行操作，不会影响当前的服务进程，从而有效提高了服务的可用性。虽然在产生新的进程和生成新的日志文件的时候，需要暂停对外的服务，以便完成日志文件的切换工作，但与 Birrell 方法相比，目录服务的中止时间要短得多。

但是，Lam 模型应用到 VO 时仍然存在一些问题。首先，该目录服务的设计是基于 Socket 的。如前文所述，在 VO 中，很多参与实体都是自治的本地域，通常会采取防火墙等保护措施以保证自身的安全，而基于 Socket 的应用很难穿越防火墙。因此，会阻碍目录服务在 VO 中的应用。其次，虽然 Lam 方法比 Birrell 的能提供更高的可用性，但平均响应时间和最坏响应时间的差距仍然很大。然而，现实应用中存在大量对服务的实时性和稳定性要求较高的系统，目录服务响应时间的振荡将对应用产生很大的影响，所以希望目录服务的响应时间应该尽可能平稳，避免出现振荡。此外，Birrell 和 Lam 的方案中都没有考虑目录服务自身的安全性，在实施 VO 访问控制时，一旦用于存储证书、密钥、访问控制策略等关键信息的目录服务器收到攻击，整个网格计算系统的访问控制将被破坏或遭到恶意的利用。

基于上述问题，本文提出了一种基于 Web Services 的轻量级高可用安全目录服务模型。该模型在可靠性、可用性和安全性方面作出了较好的平衡，可以有效支持 VO 的访问控制实施应用。

3 高可用安全目录服务

3.1 基于 Web Services 的目录服务系统架构

本文提出的目录服务系统架构如图 1 所示。在该架构中目录服务从内存文件中获取，但同时在磁盘上存贮对应的检查点和日志文件。内存数据可以通过检查点和日志来重构。显然，比起使用商品化的数据库系统来说，这种轻量级目录服务的实施和维护成本都将大大降低，因而更适宜在网格计算环境下部署。

采用这种方案，对目录数据的更新操作将逐一在硬盘的日志文件中记录，但是检查点文件却并不是在本地产生。相反，目录服务器创建一个新的进程—检查点生成进程，用来生成新的检查点，并启动新的日志文件以便记录随后的更新操作。检查点生成进程根据旧检查点文件和相关的日志记录来生成新的检查点文件。当第一个检查点文件生成时，并没有

旧的检查点文件可以获取，进程的初始化被用来获取初始化数据映像。如果在新的检查点文件完全生成前，检查点生成进程失败。那么当检查点机制在下一刻被触发时，更多的日志文件将连同旧日志文件一起，为另一个检查点生成进程使用。因为检查点被不同的进程在不同的地址空间执行，因此目录服务并不会因为检查点的生成而被延迟。

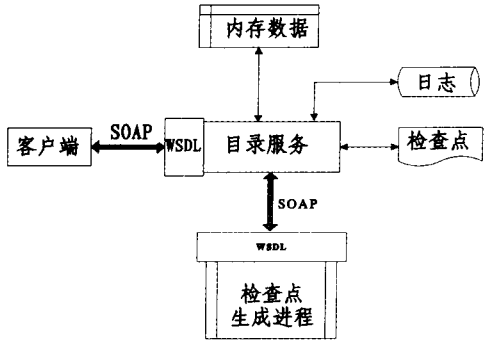


图 1　基于 Web Services 的目录服务系统架构

由图 1 可以看出，该目录服务是基于 Web Services 架构的。目录服务器进程和检查点生成进程都可以作为 Web Services，并用 WSDL 来加以描述。客户端和服务器间之间或者服务器之间的所有通信都基于 SOAP 协议。这种架构的优点不仅在于其能够满足虚拟组织聚集异构自治系统和穿越防火墙的需求，而且能够提高服务的可用性。

前面曾提到，尽管 Lam 方法比 Birrell 方法具有更高的可用性，但是在响应上仍存在延迟。这是由于检查点生成进程的初始化以及从目录服务器磁盘上读取旧的检查点文件和相关日志文件时仍然占用了大量的系统资源，造成了响应延迟。因此，平均响应时间和最坏响应时间的巨大差距在 Lam 方法中仍比较明显，这种响应时间上的振荡是基于虚拟组织的网格计算应用系统所不愿看到的。在本文的模型中，使用基于 HTTP 的 SOAP 协议进行通信。和基于 Socket 的应用相比，这种通信方式完成一次事务需要更多的时间，因此平均响应时间将有所增加。这样，平均响应时间和最坏响应时间的差距将被缩短。

此外，在实验中发现如果在另一台服务器上生成检查点进程，那么本地服务器的计算负载就会被网络和生成检查点的服务器所分担。这样的部署将使目录服务的可用性得到进一步加强。这样做的另一个好处是，如果使用低端服务器作为检查点服务器时的响应时间比用高端服务器作为检查点服务器时的响应时间短。其原因在于低端服务器占用网络带宽的能力比高端服务器低，因此必须使用更多的时间完成检查点工作，从而对目录服务器的影响就降低了。而高端服务器占用带宽的能力很强，单位时间内能够发送较多的数据包到目录服务器，以致于加重了目录服务器的负荷，降低了它对客户端的响应能力。在网格计算系统中，存在着许多较低计算能力的服务器，一般很难被用于协同计算的工作中。现在，可以充分利用它们来做检查点的工作，这是符合网格计算的基本思想的。

需要指出的是，本文提出的机制只把工作焦点放在目录服务自身的数据管理问题上。在此基础上通过一定的设计和编码工作，就很容易和一些标准目录访问协议兼容。例如，配合 DSML[9] 网关，本文的架构能够支持 LDAP 等目录服务访问协议。

3.2　基于服务迁移机制的安全扩展

为了解决目录系统的安全问题，确保目录服务顺利进行，作者在文[10]对前文提及的目录服务做了安全扩展，提出了基于代理服务器的目录服务模型。在保持目录服务的可靠性与可用性的同时，提高了其抵御外部攻击的能力。

在基于代理的目录服务中，客户通过代理向多个目录服务器发送目录服务请求，工作模式图 2 所示。在系统运行过程中，每个目录服务器通过系统广播接受来自代理的目录服务请求，并由某个目录服务器充当活动服务器（获得 Token 的服务器）。活动服务器除了接受目录服务请求外，还负责向代理回送目录服务请求应答。在每个目录服务器上应当保证目录数据的一致。

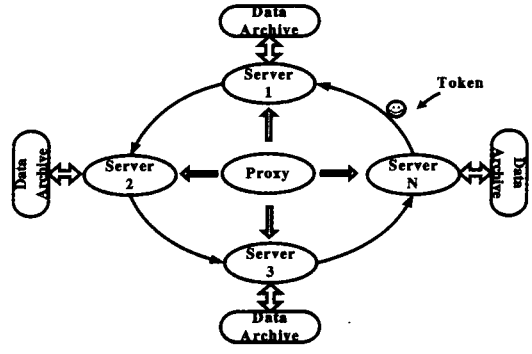


图 2　多服务器工作模式

基于代理服务器的目录服务解决方案，可以在保持目录服务工作性能的同时有效提高目录服务自身的安全性。但是，这种方案也存在着不足之处。首先，在这种架构下，尽管目录服务器组得到了保护，但代理服务器的安全必须得到保障。在实际应用中，可以通过与相应的访问控制及入侵检测系统的配合来提高代理服务器的安全级别。例如，文[11]和[12]提出的基于移动 Agent 的解决方案可以有效抵御 DoS 攻击。但是，无论采取何种保护手段，由于代理服务器成为安全瓶颈，必然要花费很大的维护代价来保护它。其次，基于代理服务器的目录服务解决方案会显著增加系统在硬件方面的实施和维护成本。对于虚拟组织来说，所期望的目录服务不仅在软件实现方面是轻量级的，在硬件构成方面也应当是轻量级的。所以，本方案适用于网格计算环境中一些极为重要的关键目录服务，而针对其它一些常规的目录服务，本文提出了基于服务迁移机制的安全扩展模型。

基于服务迁移机制的安全扩展模型如图 3 所示。

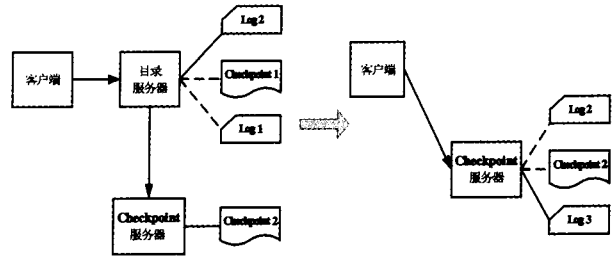


图 3　基于服务迁移机制的安全扩展模型

该模型的核心思想是当目录服务收到攻击时，系统不是被动防御，而可以采取主动迁移目录服务器的措施。从图 3 可以看出，目录服务器与归档服务器不再是固定不变的。上图左边的部分显示的是依据前面设计的目录服务的工作过程。当目录服务器准备归档时，产生新的日志文件 Log2 来记录客户端的目录操作，同时将老的归档文件 Checkpoint1 和日志文件 Log1 传送给归档服务器进行归档。归档服务器完成归档工作后，应当将归档文件 Checkpoint2 传送回目录服务器。但是，在服务迁移模型中，如果目录服务器收到攻

击，目录服务器将把当前的日志文件 Log2 传送到归档服务器，归档服务器将作为新的目录服务器并产生新的日志文件 Log3 来记录客户端操作。这样，目录服务器的迁移操作就成功完成。

显然，较代理服务器模式而言，迁移模式的实施代价和维护成本要小得多。此外，由于在迁移过程中减少了归档文件的传递次数，将进一步减小网络负载，提高系统性能。

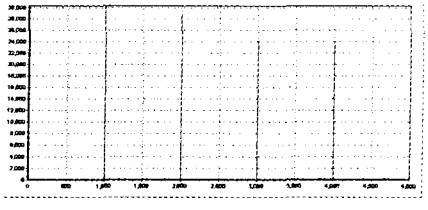
4 性能测试

4.1 基于 Web Services 的目录服务

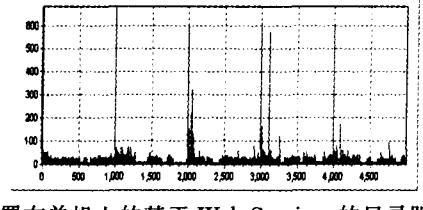
为了验证基于 Web Services 目录服务的实用性，采用不同大小的目录数据测试该试验模型。目录大小分别为 8M，16M，32M 和 64M。目录服务的性能测试在几台相同配置的 PC 机上进行。这些 PC 机的 CPU 为 Intel Celeron(1705M Hz)，内存 256M DDR，网络适配器为 Realtek8319(A)PCI 高速以太网适配器。

现针对目录大小为 8M 时的试验结果作性能评估。试验测试了客户端连续发送 5000 条请求时的响应时间。在此过程中，产生了 4 次检查点以检验它对响应时间的影响。每次成功的处理只能在目录服务成功完成前一条请求处理后才能处理本次请求，并发送一次确认。完成每条目录服务请求所花费的时间由客户端记录。试验结果图中，横坐标表示发送的请求条数，纵坐标表示响应时间(单位为毫秒)。为了清楚显示本文模型的可用性，将同时给出 Birrell、Lam、本文模型在相同试验环境下的试验结果图。

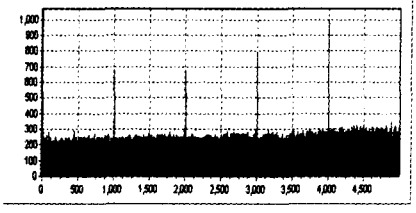
1)Birrell 目录服务



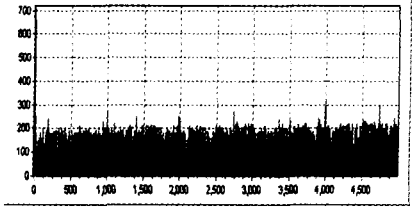
2)Lam 目录服务



3)部署在单机上的基于 Web Services 的目录服务



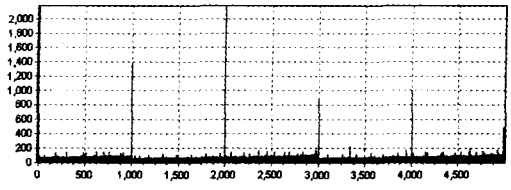
4)部署在多机上的基于 Web Services 的目录服务



从上述实验结果可以看出，本文的目录服务模型在可用性方面做出了很好的权衡，既保持了相对较低的平均响应时间，又减小了响应时间的振荡，因而更加符合虚拟组织访问控制的需要。

4.2 基于服务迁移机制的安全扩展

下图显示了基于服务迁移机制的目录服务在目录数据为 8M 的情况下，当客户端连续发送 5000 条请求时的响应时间。在此过程中，做了 4 次目录服务切换以检验它对响应时间的影响。显然，与部署在多机上的基于 Web Services 的目录服务相比，服务迁移机制能够更加有效地降低平均的响应时间。实际上，很多网格计算应用都对系统带宽相当敏感，迁移模式对它们来说是很合适的。当然，由于需要在目录服务切换时付出额外的连接代价，在服务切换时的等待时间有所延长，从而降低了系统的可用性。但是，相对于该方案在安全上给目录服务带来的好处来说，这种损失是可以接受的。



结论 本文分析了网格计算系统中目录服务的重要性，指出在虚拟组织中实施访问控制时要求目录服务是轻量级的，具备良好的可靠性、可用性和安全性，并建立通用的工作协议和系统架构之上。针对上述需求，设计了基于 Web Services 的轻量级目录服务系统架构，通过实验分析对该架构做了性能评估。此外，考虑到目录服务自身的安全，本章提出了基于服务迁移机制的目录服务安全扩展。

参 考 文 献

1 ISO. Information Technology-Open Systems Interconnection-The Directory: Abstract Service Definition. ISO/IEC 9594-3 (also ITU-T X.511), 1995

2 Yeong W, Howes T, Kille S. Lightweight directory access protocol. RFC 1777, 1995. http://www.ietf.org/rfc/rfc1777.txt

3 Wahl M, Howes T, Kille S. Lightweight directory access protocol (v3). RFC 2251, 1997

4 Hodges J, Morgan R. Lightweight directory access protocol (v3). RFC 3377, 2002

5 Lam K Y, Zhao X B, Chung S L, et al. Enhancing Grid security infrastructure to support mobile computing nodes. In: Proceedings of 4th International Workshop on Information Security Applications (WISA 2003). Lecture Notes in Computer Science, Vol 2908, Springer, Berlin, 2004. 42～54

6 Fitzgerald S, Foster I, Kesselman C. A Directory Service for Configuring High-Performance Distributed Computations. In: the Sixth IEEE International Symposium on High Performance Distributed Computing, Aug. 1997. 365～375

7 Birrell A D, Jones M B, Wobber E P. A simple and efficient implementation for small database. In: Proceedings of the 11th ACM Symposium on Operating Systems Principles. Austin, USA, 1987. 149～154

8 Lam K Y, Salkield T. Implementing a highly available network directory service. Journal of Systems Software, 1997, 37: 41～47

9 Organization for the Advancement of Structured Information Standards (OASIS). DSML V2 Specification. https://www.oasis-open.org/committees/dsml/, December 2001

10 李晖，赵曦滨，顾明，等. 基于代理的高性能目录服务的设计和实现. 计算机科学，2005(3)

11 Mell P, Marks D, Mclarnon M. A denial-of-service resistant intrusion detection architecture. Computer Networks, Special Issue on Intrusion Detection, 2000, 34: 641～658

12 Li T, Chew W M, Lam K Y. Defending against distributed denial of service attacks using resistant mobile agent architecture. In: Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS'02). Fort Lauderdale, Florida, USA, 2002