doi:10.3772/j.issn.1002-0470.2024.04.001

面向高性能计算环境的多维自适应授权访问策略①

和 荣②*** 王小宁* 肖海力* 卢莎莎* 赵一宁* 迟学斌***

(*中国科学院计算机网络信息中心 北京 100083)

(**中国科学院大学 北京 100049)

摘 要 高性能计算能力是国家综合实力和创新能力的重要体现,是支撑我国科技持续发展的关键技术之一。随着高性能计算的发展,越来越多领域的科研人员开始关注并使用高性能计算环境。高性能计算环境目前面临资源有限、用户数目增多等挑战。为保证环境的安全性、提高环境资源的利用率,需设置一定的授权访问策略来约束用户的访问行为。本文针对高性能计算环境服务对象用户和应用社区或业务平台,基于机器学习算法对用户行为进行分析获取相关属性,设计并实现了一种多维自适应授权访问策略(MAAC)。实验表明,MAAC 可实现对环境资源有效和灵活访问控制,同时该策略的决策时间可控制在 1 ms 内,与策略响应时间相比可忽略不计。

关键词 高性能计算环境;授权;属性;用户行为;安全

高性能计算能力是国家综合实力和创新能力的重要体现,是支撑我国科技持续发展的关键技术之一。自 1998 年以来,国家高性能计算基础设施建设先后得到“863 计划”、“国家重点研发计划”的持续支持[1]。由中国科学院计算机网络信息中心牵头建设的高性能计算环境(原名中国国家网格)[2]在国家科技发展中具有重要推动作用。历经十多年的发展,高性能计算环境已经接入包括国家超级计算(中国科学院)中心结点、国家超级计算天津中心结点、国家超级计算广州中心结点、国家超级计算深圳中心结点、国家超级计算无锡中心结点等在内的 22 个结点。环境提供涉及量子化学、分子模拟、高能物理、生物科学等领域的开源软件、商业软件和自主研发软件。高性能计算服务环境屏蔽了作业管理系统、接入方式、管理制度等方面的异构性,为科研人员提供了具有统一访问入口、统一使用方法和用户技术支持的高水平高性能计算应用服务。

授权是安全领域的基本元素,可确定谁可以在怎样的情况下访问特定的数据、应用和资源。高性能计算环境为保证环境本身以及资源的安全性会对其用户设置一定的授权策略。环境目前对资源的授权主要是以用户为单位,把资源划分为组,给用户分配资源组相应的权限。随着高性能计算的发展,越来越多领域的科研人员开始关注并使用高性能计算环境。然而,环境计算资源毕竟有限,用户来源越来越多样化。伴随着其他来源用户的不断接入,保证用户最大化地使用资源才是最重要的。现有授权策略为用户及应用分配权限时,无法做到根据用户的实际需求实时地改变策略。如,环境新增资源时,管理员无法智能地为已有用户增加新资源的使用权限,可能会造成资源的浪费。应用社区平台通过调用应用编程接口来使用环境资源。为保证应用的用户可以使用资源提交作业,环境会给用户分配一个默认的集群使用权限,但会出现用户并不需要使用此集群的时候,集群受账号数有限的限制会造成无法分配给更多的用户使用。因此,环境需要设计新

① 国家重点研发计划(2020YFB0204800)资助项目。

② 女,1988 年生,博士生;研究方向:高性能计算,可视化与安全技术;联系人,E-mail: herong@sccas.cn。

(收稿日期:2023-02-23)

的灵活可靠的授权策略来保护访问受限的资源。

总之,高性能计算环境授权访问面临着以下 3 个方面的挑战。

(1)用户来源和数目增加,环境需要提供灵活的访问策略保证用户方便地访问环境资源。

(2)环境资源有限,设置合理的分配策略及时释放空闲资源提高环境利用率,为更多的用户提供服务。

(3)环境需保证稳定运行,无论是用户还是应用社区平台访问环境资源时不能影响环境正常的服务运行。为此,环境需要对这些访问进行控制,包括决定是否允许访问、可访问哪些资源、限制访问流量等。

为了迎接新形势下的更多挑战,满足高性能计算的更多需求,本文提出了一种多维自适应授权访问策略(multi-dimensional adaptive access contol, MAAC)来解决高性能计算环境的授权访问问题。下面将首先介绍授权相关的技术研究;其次介绍 MAAC 具体的定义、工作流程及实现;再次通过部署测试评估策略的功能和性能,与已有访问控制进行对比分析;最后总结本文工作,并提出下一步工作的研究方向。

1 相关研究

访问控制旨在通过身份标识、身份验证和授权来允许、拒绝、限制和撤销对资源的访问。访问控制能保护机密信息不被其他未经授权的用户窃取,还可降低员工泄露数据的风险,并将基于网页的威胁拒之门外。大多数安全驱动的组织都不是手动管理权限,而是依靠身份和访问管理解决方案来实现访问控制策略。访问控制作为安全的一项重要手段,目前已有很多相关方面的研究,下面主要从访问控制策略、相关的学术研究以及访问控制平台三方面介绍目前访问控制相关的研究现状,通过对比发现其中的一些不足并在环境的访问控制策略中提出改进。

1.1 访问控制策略

传统的访问控制,包括自主访问控制(discretionary access control,DAC)、强制访问控制(mandatory access control, MAC)、基于角色的访问控制(role-based access control,RBAC),它们在不同的场景发挥着相应的作用。随着新型计算环境的出现,基于属性的访问控制(attribute-based access control, ABAC)由于有效解决了新型环境大规模、强动态性以及强隐私的特性得到了广泛的应用。表 1 对比分析了前面 4 种访问控制策略的优缺点。

1.2 相关学术研究

1.2.1 区块链应用于访问控制

文献[3]提出了一种基于区块链的域间访问控制模型,以基于属性的访问控制为基础,使用统一的属性标准描述各域的访问控制策略。使用区块链作

表 1 访问控制策略对比

名称	作用	优点	缺点
DAC	主体对客体的访问权限进行自主管理,主体决定是否将客体的全部或部分访问权限授予其他主体,可应用于许多系统环境	比较灵活,具有一定的可扩展性	安全性差,开销大,效率低,静态分配权限
MAC	系统强制主体服从访问控制规则,主体和客体都被系统分配一个固定的安全属性(安全等级),利用安全属性决定一个主体能否访问某个客体,应用于强调机密等级的应用领域如军事、金融等	增强信息的机密性,安全性较高	灵活性差,静态分配权限
RBAC	权限和角色关联,通过为主体分配适当的角色,使其能够获得相应角色的权限,应用较为广泛	简化策略管理,明确责任和授权,安全性高,灵活性、可扩展性较高	较粗粒度静态分配权限
ABAC	主体和客体的属性作为基本的决策要素,利用请求者所具有的属性集合决定是否赋予其访问权限	有效地解决动态大规模环境下的细粒度访问控制问题	策略执行依赖第三方机构

为访问控制策略的载体,利用区块链的特点通过智能合约自动化地进行策略决策,有效地保障域间访问和数据的共享安全,同时增强了用户的自主性。文献[4]提出了一种应用区块链的数据访问控制与共享模型,用属性基加密改进了现有的区块链加密方式和数据存储方式,达到了细粒度访问控制与安全共享的目的。文献[5]把区块链技术与基于属性的访问控制模型相结合,提出了基于区块链的分布式大数据访问控制机制,利用区块链事务来实现访问控制策略的分布式管理。

1.2.2　访问控制策略研究

文献[6]提出了多层级安全基于属性的访问控制,为属性设置不同级别的安全性,在存储属性数据时根据安全级别采用不同加密算法;在用户进行数据访问时通过提供相关密钥来判断用户身份并给出访问决策。文献[7]提出了混合云环境的共享资源模型,利用预留策略来最大化资源的使用以提高未使用资源的利用。文献[8]针对无中心 Web 服务环境,提出了一种基于智能合约的个性化访问控制方法,通过对用户的历史行为表现进行数据分析并给出衡量值来调整访问控制权限。文献[9]提出了基于用户体验的动态负载均衡算法(quality of experience based request fair scheduling, QRFS),保证用户满意度的前提下兼顾服务器的负载和请求调度的公平性。QRFS 算法主要对不同类型用户发起的请求设定不同的优先级,根据优先级确定请求收益值,并依此来调整请求的响应。

1.3　访问控制平台

1.3.1　亚马逊身份管理

亚马逊身份管理(identity and access management,IAM)[10]主要管理亚马逊云计算服务的服务和资源的访问,提供集中式的数字身份管理、认证授权、审计的模式和平台。IAM 提供统一的认证策略,确保高安全级别的应用程序可受到更强的认证方法保护;提供精细的权限,针对不同人员授予不同的权限。

1.3.2　阿里云资源访问管理

阿里云资源访问管理(resource access management,RAM)[11]是阿里云提供的管理用户身份与资源访问权限的服务。RAM 允许在一个账号下创建并管理多个身份,单个身份或一组身份可以分配不同的权限,从而实现不同用户拥有不同资源访问权限的目的。RAM 提供 2 种资源授权模型:云账号内授权模型和资源组内授权模型。

1.3.3　华为云应用身份管理服务

华为云应用身份管理服务 OneAccess[12]是华为云提供的应用身份管理服务,具备集中式的身份管理、认证和授权能力,保证企业用户根据权限访问受信任的云端和本地应用系统,并对异常访问行为进行有效防范。OneAccess 支持根据用户的工作智能定义权限的粗粒度授权机制,同时也可以提供精细到具体服务的操作、资源以及请求条件等的细粒度授权。多数细粒度策略以 API 接口为粒度进行权限拆分,权限的最小粒度为 API 授权项。

表 2 从功能、访问控制策略、控制粒度 3 个方面对上述的访问控制平台进行了对比分析。由此可以看出,每一个访问控制平台在具备访问控制的同时也具备身份管理的功能;在访问控制策略以及粒度上都支持通过角色、资源来定义策略,同时针对细粒度的服务都有对应的策略。

表 2　访问控制平台对比

名称	功能	访问控制策略	控制粒度
亚马逊	身份管理 认证授权 审计	基于身份的策略 基于资源的策略 访问控制列表 会话策略	用户 角色 服务
阿里云	身份管理 资源访问	基于角色的策略 基于身份的策略 基于资源的策略	用户 用户组 角色 服务
华为云	身份管理 智能访问 控制	基于角色的策略 基于资源的策略 基于身份的策略	用户 服务 资源

目前关于访问控制的相关学术研究基本上集中在利用区块链技术来保证系统的安全性、分析用户的属性动态调整权限以及属性加密机制上。本文后面提出的策略会借助机器学习算法来分析用户历史行为,根据用户的属性来动态调整用户权限。

2 多维自适应授权访问策略

高性能计算环境包含 2 种工作模式:用户通过命令行或环境通用计算平台[13]直接访问环境资源;第三方应用通过环境应用编程接口[14]访问环境资源。本文从环境的服务对象、用户的来源、角色以及属性等方面考虑设计了环境的授权策略,最终采用多维自适应授权访问策略 MAAC 来限制外部的访问请求,保证环境资源的安全。

2.1 策略模型定义

为了方便理解后面介绍的相关策略和流程图,表 3 介绍了环境服务对象相关的一些概念。

表 3 环境服务对象相关概念

概念	说明
环境账号	环境认证系统的网格账号
第三方应用	通过接口支持的应用社区或业务平台
第三方应用账号	应用社区或业务平台的用户,通过自身的认证系统进行身份认证
用户来源	用户通过哪种认证途径来访问资源,取值为环境账号或第三方应用账号
角色	根据用户功能划分为三类角色:普通用户、技术支持者、管理员
用户属性	用户常用的应用、集群以及队列信息

2.1.1 用户授权策略

用户授权模型中用到的相关概念、具体的含义以及表示方式如表 4 所示。模型中的权限主体为用户,用户具有来源、角色以及相关的属性特征,这些特征决定用户最终的权限;客体为要访问的资源,包括集群、队列、应用以及作业。

表 4 中主体对客体的操作具体语义参照表 5。

通过此模型可清楚方便地为用户定义相应的授权策略。下面是一个用户授权策略的示例:

$U=\{U1\}$,其中 $U1=(test,env,normal,cluster1)$ 表示环境账号用户 *test*,所属角色为普通用户,且具有集群 *cluster*1 属性。

$O=\{cluster1,queue1,app1\}$ 表示要访问集群 *cluster*1 上 *queue*1 队列的 *app*1 应用。

$P=\{S\}$ 表示对上述定义的资源要进行使用的操作。

$X=(\{(test,env,normal,cluster1)\},\{cluster1,queue1,app1\},\{S\})$ 表示用户 *test* 对集群 *cluster*1 上 *queue*1 队列的 *app*1 应用具有使用的权限。

表 4 用户授权模型相关概念

概念	说明
主体	获得策略中定义的权限主体,这里主要指用户,采用 $U=\{U1,U2,U3,\cdots\}$ 表示,其中:$Ui=(name,group,role,attr1,attr2,\cdots,attrn)$ *group* 取值为 *env*(环境账号)或者 *third*(第三方应用账号) *role* 为用户所属角色,取值为 *normal*(普通用户)、*supporter*(技术支持者)和 *admin*(管理员) *attri* 为用户的属性,可能是集群、应用或队列
客体	要访问的资源,这里指集群、队列、应用、作业,采用 $O=\{C,Q,A,F\}$ 表示,其中: $Ci=(name,attr,a1,a2,\cdots an,q1,q2,\cdots qm)$, Ai 属于 A,Qi 属于 Q $Ai=(name,attr,q1,q2,\cdots qn)$, Qi 属于 Q
操作	主体对客体进行的访问操作,这里主要包括管理、读取和使用 3 种操作,采用 $P=\{M、R、S\}$ 表示
权限	主体对客体某种操作的权限,采用 $X=(u,o,p)$ 表示用户 u 对资源 o 具备 p 的权限

表 5 用户授权模型操作语义

资源类型	管理 M	读取 R	使用 S
集群 C	修改集群配置信息	读取集群信息	-
队列 Q	修改队列配置信息 维护 S 的授权信息	读取队列信息 读取 S 的授权信息	往该队列提交作业
应用 A	修改应用配置信息 维护 S 的授权信息	读取应用配置信息 读取 S 的授权信息	提交该应用作业
作业 F	调整作业状态信息 终止异常作业	读取作业基本信息 读取/下载文件	-

2.1.2 第三方应用授权策略

第三方应用通过编程接口来访问环境,所以对应用提供可访问一组或多组编程接口的权限,具体模型如下所述。

表 6 介绍了第三方应用授权模型的相关定义,与用户授权策略不同之处在于面向的主体是第三方应用,客体表示一组应用编程接口。下面是一个第三方应用授权策略的示例:

$U=\{U1\}$,$U1=(thirdappid)$ 表示第三方应用

thirdappid

$O = \{jobNormal\}$ 表示要访问 *jobNormal* 组的接口

$P = \{get\}$ 表示对上述定义的资源要进行 *get* 的操作

$X = (\{(thirdappid)\},\{jobNormal\},\{get\})$ 表示第三方应用 *thirdappid* 对属于 *jobNormal* 组的接口具有 *get* 权限。

表 6　第三方应用授权模型相关定义

概念	说明
主体	获得策略中定义的权限主体,这里主要指第三方应用,采用 $U = \{U1,U2,U3,\cdots\}$ 表示,$Ui = (appid)$
客体	要访问的资源,这里指一组接口,采用 $O = \{I\}$ 表示
操作	主体对客体进行的访问操作,这里主要包括 *get*,*put*,*post*,*delete* 4 种操作,采用 $P = \{get,put,post,delete\}$ 表示
权限	主体对客体某种操作的权限,采用 $X = (u,o,p)$ 表示应用 u 对属于 o 组的接口具有 p 权限

2.2　策略设计

针对环境服务对象的不同分别设计了用户授权访问策略和第三方应用授权策略。针对用户授权访问,本文考虑了从角色和用户属性 2 个维度进行访问控制,动态调整用户的授权策略;考虑第三方应用通过调用应用编程接口来访问环境资源,从接口权限分配和访问请求控制 2 个方面来限制第三方应用的访问。

2.2.1　基于角色的授权访问

环境根据用户访问资源的不同设置 3 类角色:普通用户、技术支持者和管理员。环境普通用户具备用户访问环境资源的基本权限,包括读取作业、应用、集群以及队列,使用集群、队列提交作业;环境技术支持者为用户提供技术支持,除具备普通用户权限外,可读取所有用户的作业、作业文件以及下载作业文件;环境管理员可访问所有资源,包括修改应用、队列以及集群的配置信息、维护授权信息、调整作业状态等。

图 1 表示基于角色的授权访问流程,获取用户所属角色后根据角色对应的权限进行相应操作。

2.2.2　基于用户属性的授权访问

用户属性初始值为申请账号时指定的常用应用信息。后续利用机器学习算法分析用户日志及作业信息,归纳用户常用集群及应用信息,动态调整用户属性。

图 2 表示用户属性的提取流程,具体每个阶段如下。

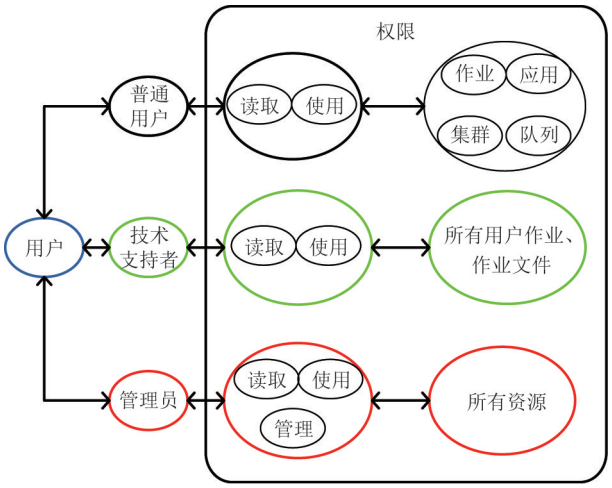

图 1　基于角色的授权访问

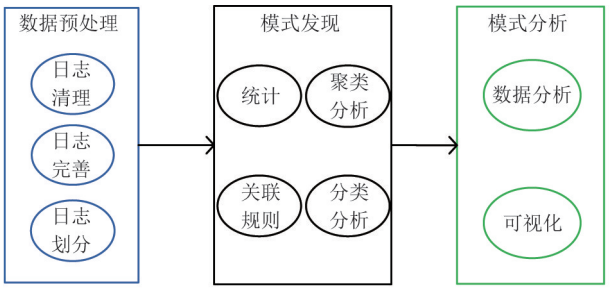

图 2　用户属性提取流程图

(1)数据预处理

系统中用户的访问原始日志记录并不适于直接分析,必须进行适当的预处理。预处理操作主要包括:日志清理,去除无用的记录;日志完善,补充某些记录访问路径为完整的访问序列;日志划分:根据用户会话划分成多个事务。

(2)模式发现

模式发现是对预处理后的数据用机器学习算法来分析,主要包括统计、聚类分析、关联规则以及分类分析等算法。

1)统计

分析会话文件,对浏览时间和浏览路径等进行频度、平均值等统计分析。

2)聚类分析

从日志数据中聚类出具有相似特性的用户。通过聚类可发掘用户常用的应用以及集群,方便后续进一步调整权限。

3)关联规则

寻找在同一个事件中出现的不同项之间的相关性。

4)分类分析

找出定义了一个项或事件是否属于数据中某特定子集或类的规则,常用决策树方法类来进行分类。

(3)模式分析

对原始数据进行进一步分析,找出用户的访问规律,即用户的兴趣爱好及习惯,并使其可视化。

2.2.3　接口权限分配

按照以往第三方应用调用接口的情况,目前接口分组情况如表 7 所示。默认情况为第三方应用分配 *jobNormal* 组权限,如有特殊需求可申请分配其他分组接口。应用中的用户根据申请时的授权分配资源读取权限和默认队列使用权限,后期根据使用情况归纳用户属性,分配其他资源使用权限。

表 7　接口分组情况

分组名称	描述
jobNormal	作业普通权限,包括查看资源、上传文件、提交作业、查看作业及作业文件、下载结果
jobAdmin	管理其他用户作业
viewNormal	查看作业统计信息相关内容
queryNormal	查询环境整体信息的普通权限
userAdmin	管理用户的申请、创建和映射等

2.2.4　访问请求控制

根据以往接口运行经验,设置一定的访问请求控制策略实现请求的最大化满足,以最快的速度对用户请求做出响应。

(1)用户流量限制

对用户单位时间内的调用次数设定最大访问值。避免用户频繁调用接口,设置用户每小时访问接口的最大值,每访问一次接口,访问次数累计,达到最大之后直接拒绝用户的请求。

(2)接口流量限制

对于作业提交接口,为避免用户批量提交作业,同一时间限制用户提交作业数目;对于需要与核心服务交互才能获取数据的接口,受网络安全协议最大连接数的限制,对接口设定最大访问次数。

(3)应用流量限制

对调用接口的应用社区或业务平台设置一个总的最大访问次数,避免应用被攻击后持续访问接口,造成接口服务无法正常服务。

(4)其他措施

对于统计类的接口,借助缓存等服务来存储数据,避免接口一直与后台服务交互。后台接口服务部署时采用微服务的方式部署,避免相互之间的影响,对访问可做到分流限制。

2.3　策略实现

2.3.1　授权访问流程

图 3 是完整的授权访问流程图,具体流程如下所述。

(1)用户来源判断

用户发送访问请求,获取请求中用户来源字段,判断用户是否为环境账号,进而决定通过哪些访问策略为用户请求作出响应。

(2)策略判定

环境账号发送的请求进行用户授权策略判定,包括基于角色的授权访问和基于用户属性的授权访问。

第三方应用账号的请求进行第三应用授权策略判定,满足条件后进行基于用户属性的授权访问,满足条件则允许第三方请求访问,否则拒绝访问。

(3)合并判定结果

环境账号发送的请求需经过合并判定结果才能获取最终响应。所有策略都满足时允许用户的访问请求,其余情况拒绝访问。

2.3.2　用户属性获取

目前已有很多现成的 Python 库可实现日志数据的分析。这里选用 Pandas[15] 来对数据进行提取、聚类分析进而获取用户属性。

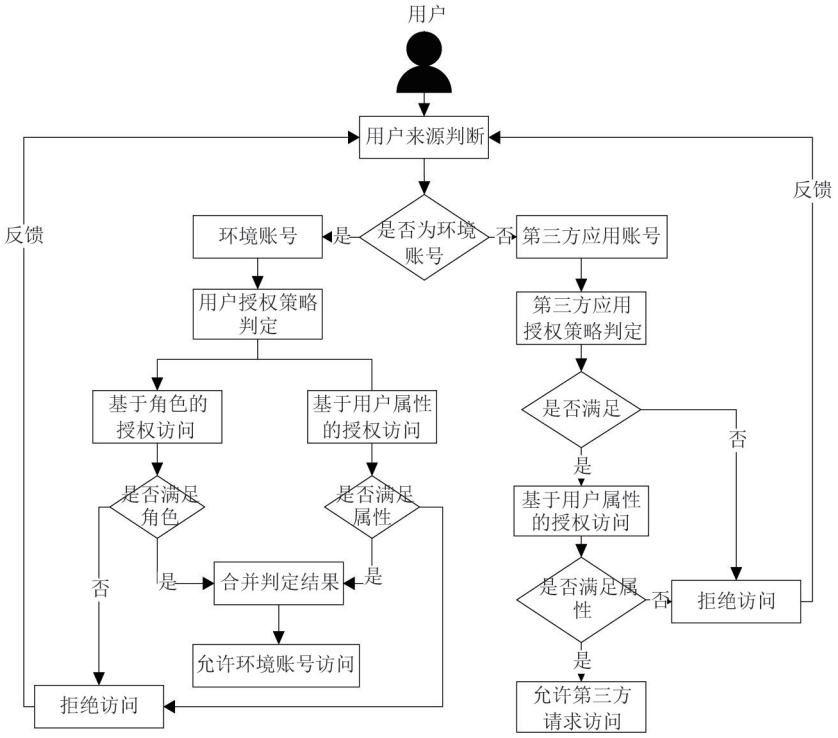

图 3　授权访问流程图

鉴于采用哪种算法处理数据并不是本文研究的重点,只要能够实现提取出用户属性即可,后续会对相关算法进行对比分析,采用更加高效准确的算法。

2.3.3　用户授权访问

用户授权访问通过权限检测模块和权限管理模块来实现。权限检测模块实现对用户授权策略的判定,每个用户的请求都需经过权限检测模块进行权限判断;权限管理模块负责维护权限策略,增加、删除或更新其中的策略。

2.3.4　接口网关

第三方应用的授权访问通过接口网关来实现,具体结构图如图 4 所示。网关接入授权控制策略,请求到达时需完成以下 4 类操作。

(1)身份认证检测

确保用户身份信息的有效性。

(2)接口访问权限管理与鉴别

判断应用及用户是否具有访问该接口的权限。

(3)参数合法性检测和接口安全防护

保护后端系统和数据不被恶意侵犯和篡改。合法性主要是检测用户请求路径或参数是否有错误;对接口调用进行完整性、有效性、唯一性鉴定,包括对请求参数加密、参数携带的时间戳核对。

(4)流量监控与管理

对最大连接数、HTTP 连接以及每秒请求数的限制。设置一定的流量控制策略实现请求的最大化满足,以尽快的速度对用户请求作出响应。

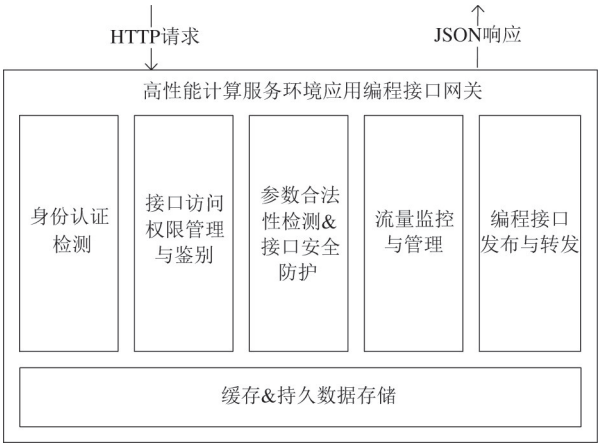

图 4　高性能计算环境应用编程接口网关结构图

3　部署测试

为对 MAAC 进行更好地评估,下面从功能和性能 2 个方面对策略进行测试。测试表明MAAC与

其他策略相比在保证环境的稳定运行上确实更加灵活,同时响应时间等性能指标都不低于原有访问策略。

3.1　实验环境

为更好地控制和测试策略的相关指标,本文把环境相关服务以 Docker 容器的形式部署在 Linux 服务器上,包括接口网关服务、权限检测和权限管理模块等。服务器详细配置如下:64 位 CentOS 6.05 操作系统,8 GB 内存,Intel Core Processor (Skylake)处理器。

实验环境配置详细信息见表 8,包括集群、部署安装的应用软件、集群设置的不同队列信息和用户列表。根据测试需求为用户分别配置不同的属性、设置不同的权限策略。如 *test*1 用户可以往集群 *cluster*1 上 *queue*1 队列提交 *app*2 应用,但不能往集群 *cluster*1 上 *queue*2 队列提交 *app*2 应用。

表 8　测试环境配置

集群	应用软件	队列	用户列表
*cluster*1	*app*1	*queue*1	*test*3
		*queue*2	*test*2
	*app*2	*queue*1	*test*1
		*queue*2	-
*cluster*2	*app*1	*queue*1	*test*2,*test*3
		*queue*2	-
	*app*2	*queue*1	-
		*queue*2	*test*1
*cluster*3	*app*1	*queue*1	*test*2
		*queue*2	*tets*3
	*app*2	*queue*1	-
		*queue*2	-

3.2　功能测试

3.2.1　正确性

从不同的访问请求能否给出正确响应以及多次请求正确响应率来反馈策略的正确性。正确响应率 r 计算如式(1)所示。

$$r = E/T \tag{1}$$

其中 E 代表返回正确响应的请求次数,T 代表请求总数。

(1)用户策略测试

通过命令行测试用户是否允许访问并使用环境资源,具体作业提交命令为

bsub-h *cluster*-q *queue appname*　　(2)

其中,-h *cluster* 表示所采用的集群名称;-q *queue* 表示队列名称;*appname* 表示提交的应用软件名称。

系统中针对用户 *test*1 的一条授权策略为

({*test*1,*env*,*normal*,*app*2}, {*cluster*1,*queue*1, *app*2},{*S*,*R*})　　(3)

表 9 表示在上述策略下用户通过改变队列、应用以及集群发送不同请求得到的响应。由响应结果可以看出,本文设定的策略确实能够实现限制用户访问资源的功能,且每次都能得到正确的响应。

表 9　用户请求响应表

测试内容	请求	响应	说明
队列属性验证	bsub-h *cluster*1-q *queue*1 *app*2	正常返回结果	使用队列 *queue*1 时返回正确结果;
	bsub-h *cluster*1-q *queue*2 *app*2	拒绝访问	使用队列 *queue*2 时拒绝访问
应用属性验证	bsub-h *cluster*1-q *queue*1 *app*1	拒绝访问	提交 *app*1 应用时拒绝访问;
	bsub-h *cluster*1-q *queue*1 *app*2	正常返回结果	提交 *app*2 应用时返回正确结果
集群属性验证	bsub-h *cluster*1-q *queue*1 *app*2	正常返回结果	使用集群 *cluster*1 时返回正确结果;
	bsub-h *cluster*2-q *queue*1 *app*2	拒绝访问	使用集群 *cluster*2 时拒绝访问

为更精确地测试请求的正确响应率,随机发送表 9 中的不同请求。图 5 显示不同请求次数下的正确响应率,可以看出尽管请求总数增多,但正确响应率依然可维持在 100%。

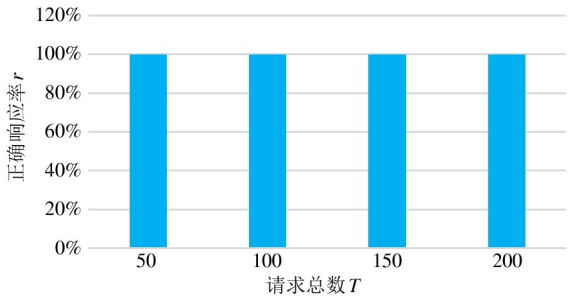

图 5　不同请求次数下的正确响应率

(2)第三方应用策略测试

第三方应用 *portal* 通过编程接口来使用环境资

源,环境为其设置的授权策略列表为

$$(\{(portal)\},\{jobNormal\},\{get\}) \tag{4}$$

为方便测试请求最终是否得到正确响应,用户授权策略列表中增加 1 条如式(5)的权限。

$$(\{(puser1,third,normal,cluster1)\},\{cluster1,queue1,app2\},\{R\}) \tag{5}$$

其中, *puser*1 为 *portal* 用户。

表 10 所列为环境收到用户 *puser*1 通过第三方应用调用不同的编程接口后给出的响应,根据响应结果可以看出本文设定的策略每次都能得到正确的响应。

表 10　第三方应用请求响应表

请求	响应	说明
get /resources/applications	正常返回结果	-
get /resources/hpcs	正常返回结果	-
post /jobs	拒绝访问	没有 *post* 权限
get /jobs	正常返回结果	-
get /data/jobs/52/cs	正常返回结果	-
put /data/jobs/52/cs/filename	拒绝访问	没有 *put* 权限

3.2.2　灵活性

设计 MAAC 的目标之一是根据用户的实际需求实时地改变策略,提高用户授权访问的灵活性。下面用具体的场景来说明。

(1)用户 *U*1 长期提交应用 *A*1 的作业,所以其属性列表中有一条 *A*1 的选项。当环境新接入的集群 *C*1 部署有应用 *A*1 时,通过在权限列表中为用户增加一条权限规则:$(\{U1,env,normal,A1\},\{C1,A1\},\{S\})$ 即可实现用户 *U*1 可使用集群 *C*1 的应用 *A*1 权限。

(2)来源于第三方应用的用户默认具有集群 *C*2 的使用权限,经过一段时间运行后发现用户并没有使用集群 *C*2 的日志。为提高集群的映射率,通过取消用户的属性 *C*2,实现不再让用户具备使用集群 *C*2 的权限。

(3)环境监控到集群 *C*3 的利用率比较低时,可选取环境活跃用户为其增加使用集群 *C*3 的权限。

针对以上场景,原始访问控制策略并不能实现实时地改变用户的权限,同时可能造成资源的浪费。相比之下,MAAC 可灵活地满足各个场景的需求。

3.3　性能测试

表 10 中 6 个请求分别代表了查询应用、查询集群、提交作业、查询作业、查询作业列表及下载文件 6 类基本的操作。因此,本文采用这 6 个请求对 MAAC 性能进行测试,同时与环境原有访问策略进行对比。

图 6 给出了不同请求在 MAAC 与原始访问策略下需要的响应时间。由图可以看出,MAAC 对请求响应产生的性能开销与原始访问策略相差不大,大部分请求优于原始访问策略。

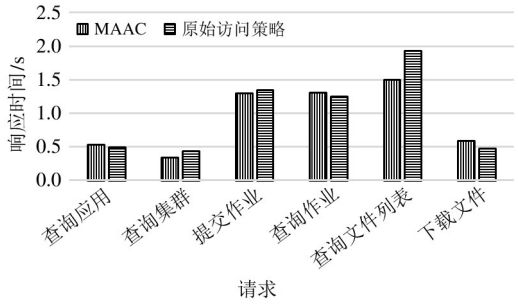

图 6　MAAC 性能测试

图 7 给出了 6 个请求对应的 MAAC 策略判定时间,可以看出,所有请求的判定时间都不超过 1 ms,与请求的整体响应时间相比可忽略不计。

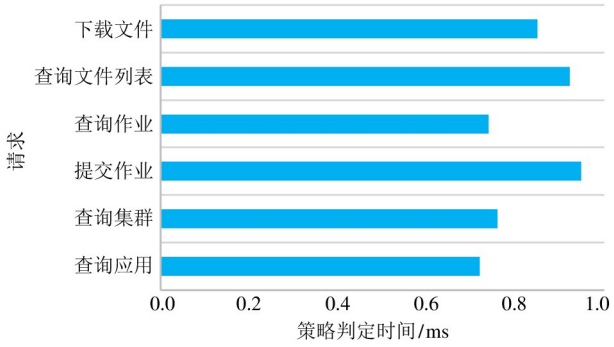

图 7　不同请求下的 MAAC 策略判定时间

4　对比分析

通过对比分析来评估本文提出的授权访问策略。将本文策略与 1.1 节提出的访问控制策略从海量性、动态性、分布式的角度进行对比,具体结果如表 11 所示。可以看出,本文提出的策略结合了传统

访问控制 RBAC 和 ABAC 的特色,从多维度实现了访问控制,在满足高性能计算环境的需求(动态调整权限、用户数目增多、用户分布式存储)上具有一定的优势。

表 11 MACC 与传统访问控制策略对比分析

策略	海量性	动态性	分布式
传统访问控制	随着数据的增长,访问控制策略呈指数增长,系统开销极高且访问效率低下	静态分配权限,无法满足动态需求;粗粒度无法应对数据频繁变动	无法支持各域统一访问控制策略标准,不便于信息共享
ABAC	随着数据的增长,访问控制策略成线性增长,系统开销较小且访问效率较高	通过属性可实现动态调整权限	支持各域统一访问策略标准
MAAC	随着数据的增长,系统开销较小且访问效率较高	通过属性实现动态调整权限;细粒度可以及时管理数据	细粒度和灵活性,支持各域统一访问控制策略标准,方便信息共享

5 结 论

本文介绍了高性能计算环境目前授权访问的局限,分析了当前访问控制相关的研究,提出了多维自适应授权访问策略 MAAC。根据用户的需求构建访问控制策略,对第三方应用实现严格的访问控制,保证环境资源的安全性;另外利用机器学习算法分析用户行为获取相关属性,并通过属性动态调整用户权限,有效提高了资源的利用。目前 MAAC 已经运行在高性能计算环境中,测试结果表明该策略可有效地解决用户的授权问题,提供灵活可靠的服务。

本文提出的授权访问策略目前在实现时还存在一些尚未考虑到的问题,如用户属性获取算法、策略提高利用率的验证等。在后续工作中计划对相关算法进行对比分析,采用更加高效准确的算法加以优化完善,以确保用户充分利用环境资源的同时保证环境的安全。

参考文献

[1] 中国国家网格. 国家高性能计算环境发展报告[EB/OL]. [2023-1-20]. http://www.cngrid.org/yjcg/fzbg.

[2] XU Z, CHI X, XIAO N. High-performance computing environment: a review of twenty years of experiments in China [J]. National Science Review, 2016,3(1):36-48.

[3] 张建标,张兆乾,徐万山,等. 一种基于区块链的域间访问控制模型[J]. 软件学报, 2021,32(5):1547-1564.

[4] 王秀利,江晓舟,李洋. 应用区块链的数据访问控制与共享模型[J]. 软件学报, 2019,30(6):1661-1669.

[5] 刘敖迪,杜学绘,王娜,等. 基于区块链的大数据访问控制机制[J]. 软件学报, 2019,30(9):2636-2654.

[6] SFA A, MS A, DS A, et al. MLS-ABAC: efficient multi-level security attribute-based access control scheme [J]. Future Generation Computer Systems, 2022,131:75-90.

[7] DHANALAKSHMI B K, SRIKANTAIAH K C, VENUGOPAL K R. Carry forward and access control for unused resources in multi sharing system of hybrid cloud [J]. Future Generation Computer Systems, 2020,110:282-290.

[8] 胡斌,赵晓芳,宋永浩,等. 基于合约的 Web 服务个性化访问控制方法[J]. 高技术通讯, 2021,31(9):901-909.

[9] 郑晓辉,史骁,金岩,等. 面向用户体验的动态负载均衡算法研究[J]. 高技术通讯, 2021,31(4):359-366.

[10] AMAZON. IAM[EB/OL]. [2023-1-20]. https://docs.aws.amazon.com/IAM/latest/UserGuide/introduction.html.

[11] ALIBABA. 阿里云 RAM[EB/OL]. [2023-1-20]. https://www.alibabacloud.com/help/zh/resource-access-management/latest/what-is-ram.

[12] 华为. 华为云应用身份管理服务 OneAccess[EB/OL]. [2023-1-20]. https://support.huaweicloud.com/oneaccess/index.html.

[13] 和荣,王小宁,卢莎莎,等. 高性能计算环境通用计算平台[J]. 计算机系统应用, 2019,28(12):55-62

[14] 和荣,肖海力,王小宁,等. 高性能计算服务环境应用编程接口[J]. 计算机系统应用, 2022,31(8):184-191.

[15] PYDATA. Pandas user guide [EB/OL]. [2023-1-20]. https://pandas.pydata.org/docs/user_guide/index.html#user-guide.

Multi-dimensional adaptive access contol for high-performance computing environment

HE Rong[*,**], WANG Xiaoning[*], XIAO Haili[*], LU Shasha[*], ZHAO Yining[*], CHI Xuebin[*,**]

([*]Computer Network Information Center, Chinese Academy of Sciences, Beijing 100083)

([**]University of Chinese Academy of Sciences, Beijing 100049)

Abstract

High-performance computing (HPC) capability is an important manifestation of a country's comprehensive strength and innovation capability, and is one of the key technologies supporting the sustainable development of science and technology in China. With the development of high-performance computing, more and more researchers in the field have started to pay attention to the HPC environment. Now the HPC environment is facing challenges such as limited resources and increasing number of accounts. In order to ensure the security of the environment and improve the utilization of the environment resources, certain authorized access policies need to be set to constrain the access behavior of users. In this paper, a multi-dimensional adaptive access control (MAAC) policy is designed and implemented. The policy is based on machine learning algorithms to analyze user behavior and obtain relevant attributes for users and application communities or business platforms, which are served by the HPC environment. Experimental results show that MAAC can achieve effective and flexible access control to environmental resources. Meanwhile the determination time of the MAAC can be controlled within 1 ms, which is negligible compared with the response time.

Key words: high-performance computing environment, authorization, attributes, user behavior, security