CN43-1258/ TP
ISSN 1007-130X
计算机工程与科学
COMPUTER ENGINEERING & SCIENCE
2007 年第 29 卷第 3 期
Vol. 29 ,No. 3 ,2007

文章编号:1007-130X(2007)003-0117-02

基于 LDAP 的对象存储系统元数据的组织与管理*

LDAP-Based Organization and Management of the Metadata in Object-Based Storage Systems

王 芳,张顺达,冯 丹,曾令仿
WANG Fang, ZHANG Shun-da, FENG Dan, ZENG Ling-fang
(华中科技大学计算机科学与技术学院信息存储系统教育部重点实验室,湖北 武汉 430074)
(National Laboratory of Storage System, Huazhong University of Science and Technology, Wuhan 430074, China)

摘 要:本文介绍了面向对象系统中元数据服务器的设计及元数据的组织和管理。该系统中元数据服务器使用了轻量级目录访问协议(LDAP)作为存放元数据的平台,针对这个平台设计了相应的数据分配算法和数据转换模块,并对其性能进行了分析和优化。

Abstract: This paper introduces the design of a metadata server in object-based storage systems and the organization and management of metadata. The metadata server in our object-based storage system uses the Lightweight Directory Access Protocol (LDAP) to store metadata. And we design data allocation and data conversion modules especially for the platform. We also analyze and optimize the performance of the metadata.

关键词:对象存储系统;元数据管理;轻量级的目录访问协议
Key words: object-based storage system; metadata management; LDAP
中图分类号: TP333
文献标识码: A

1 引言

在以往的 DAS、NAS、SAN 中,由于块设备不能管理元数据,使得服务器/元数据服务器成为瓶颈。而在基于对象存储(OBS)系统中,由于与块/扇区有关的元数据管理(大约有 90 %的负载)已交由 OSD 负责,元数据服务器只管理与文件目录有关的元数据(10 %的负载),使系统瓶颈得到极大的缓解。良好的元数据管理将减少不必要的开销,加快访问速度,均衡负载,提高系统性能。

对象存储模型显现出分布式存储系统的体系结构特征。对象存储系统由客户端、元数据服务器和对象设备三大部分组成,它采用三方通讯方式提供存储服务,即客户端向元数据服务器发送读写文件请求,元数据服务器返回文件对应的对象信息和权限,客户端根据对象信息以赋予的权限访问相应的设备。传统系统中元数据和数据在同一台设备、同一个机器和同一个文件系统中[1]。基于对效率的考虑,元数据通常被单独存放在距离其描述的数据较近的物理位置上[2]。在一些分布式文件系统中,数据被存放在可以通过网络直接访问的智能设备上,而元数据由专门的元数据服务器管理[3]。我们不难发现,元数据服务器在对象存储系统中的位置非常重要,是整个系统潜在的瓶颈。元数据服务器的中心任务就是元数据的组织和管理。

2 元数据服务器的设计

2.1 元数据服务器的功能分析

对象存储系统中,元数据服务器的主要任务是完成元数据管理功能,如文件目录结构维持、用户管理、命名、权限管理,通过文件系统调用向用户提供与传统文件系统相同的文件服务。主要的关键技术可描述为:

(1)元数据的存储、访问和管理:将元数据存放到轻量级目录访问协议(LDAP)后台数据库 BDB 中,通过 LDAP 协议 API 访问数据库,方便高效地对元数据进行存储、访问和管理。由于 LDAP 针对目录访问进行了优化,在元数据管理方面具有一定优势。加之封装了较多常用功能,使

* 收稿日期:2005-12-26;修订日期:2006-03-22
基金项目:国家 973 计划资助项目(2004CB318201);国家自然科学基金资助项目(60303032)
作者简介:王芳(1972-),女,河北深县人,副教授,研究方向为计算机存储技术与网络存储系统。
通讯地址:430074 湖北省武汉市华中科技大学计算机科学与技术学院;Tel:(027)87557649;E-mail:zhangshunda@126.com
Address: School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, Hubei 430074, P. R. China

开发和维护更加方便。其可扩展性和安全性也十分优异。

(2)缓冲区管理:针对元数据访问的特点,在系统中特制一个 Buffer Cache,将最近访问过的元数据缓存到缓冲区中,以便下次访问命中时直接从缓冲区中取走数据,不必再产生磁盘 I/O。缓冲区使用类似 Hash 链表加双向链表的结构,使用 LRU 算法淘汰过时数据,提高内存利用率。

(3)文件到对象的映射:映射通过策略库调出一定的算法,根据 OSD 的负载情况、空闲空间的大小、文件本身的特点、QOS 的要求等因素决定对象 ID 的分配。

(4)对象在 OSD 间的分布:根据文件的大小决定如何放置到具体的 OSD 设备中。大文件(超过一定阈值,如 1M、2M 等)就采用分片的方式,即将文件平均分成若干份,散列在 OSD 中;小文件使用 Hash 的方法,将文件全路径名哈希成一个整型数,根据这个数字确定放置对象的设备。分片和哈希各有自身的优势,将它们分别应用到大文件和小文件中可以最大限度地发挥算法的特点,使系统性能达到最优。

2.2 元数据服务器的软件组成

元数据服务器由文件管理器、资源管理器和轻量级目录访问协议(LDAP)数据库三大部分组成。文件管理器主要负责与客户端的通信、对象分配、元数据管理;资源管理模块主要负责与设备通信、管理设备和用户、设备间负责均衡和调度;LDAP 数据库存放元数据信息和设备信息,也是文件管理模块和设备管理模块的公用平台。

与元数据组织和管理关系较为密切的是文件管理器,其具体的结构如图 1 所示。

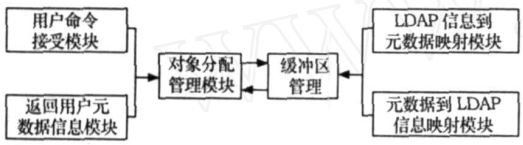

图 1 文件管理器各模块

文件管理器各主要模块包括:

(1)客户命令接受模块。负责从网络接受客户发送的读写请求。

(2)对象分配、管理模块。负责从客户命令接受模块中提取信息,根据相关信息结合 LDAP 中的信息分配对象和管理元数据。在现阶段具体就是通过文件类型和大小的判断,决定文件到对象的映射。大文件使用分片算法,小文件使用哈希算法。

(3)返回客户数据信息模块。将预分配好的对象信息封装成标准的 OSD 命令,通过网络传送给客户。

(4)元数据到 LDAP 信息映射模块。将新建的对象信息转换成 LDAP 标准格式,并提交到 LDAP 后台数据库中。比如,文件 /test/usr/src/prog.c 对应的对象会被存放到类似 dn:cn=prog.c,cn=src,cn=usr,cn=test,cn=root,dc=example,dc=com 的记录中去。而具体的对象数据结构会被存放到此条记录下的属性项中去,如 cn、sn 和自定义的属性等。再通过如 ldap_open()、ldap_bind()、ldap_add()等 API 函数向后台数据库提交结果。

(5)LDAP 信息到元数据映射模块。是元数据到 LDAP 信息映射模块的逆过程,供对象分配、管理模块调用 LDAP 信息时使用。

(6)缓冲区管理模块。当用户发送的信息到达服务器端时,首先将所得到的文件名哈希成一个整形数,以这个数为下标找到对应的数组元素。再以这个数组元素为指针找到元数据信息。如果其中的文件名与用户发送的文件名吻合,说明就是用户要找的元数据,则可直接取走;否则顺着链表向后找。若找到则取走元数据,若找不到则访问 LDAP 数据库,并将返回的元数据填充至链表头。此时,如果链表节点数达到设定的最大值,则将最后链表末尾项丢弃。这样就简单地实现了 LRU 算法。

2.3 元数据的存放和转换

元数据存放在服务器后台挂接的 LDAP 服务器中。LDAP(Lightweight Directory Access Protocol,简称 LDAP)是基于 X.500 标准的,但它简单多了,并且可以根据需要定制。与 X.500 不同,LDAP 支持 TCP/IP,这对访问因特网是必须的。就像 Sybase、Oracle、Informix 或 Microsoft 的数据库管理系统(DBMS)是用于处理查询和更新关系型数据库那样,LDAP 服务器也是用来处理查询和更新 LDAP 目录的。换句话来说,LDAP 目录也是一种类型的数据库,但不是关系型数据库。LDAP 最大的优势是:可以在任何计算机平台上用很容易获得且数目不断增加的 LDAP 的客户端程序访问 LDAP 目录,而且也很容易定制应用程序为其加上 LDAP 的支持。LDAP 协议是跨平台的和标准的协议,因此应用程序就不必为 LDAP 目录放在什么样的服务器上操心。

Linux ext2 文件系统的最大容量为 4TB[4],而 LDAP 通过多台服务级联可以达到 PB 级容量。对象存储系统中的元数据不包括块信息,大部分元数据读多写少,长度固定且较小,适合数据库特别是 LDAP 数据库模型。另外,LDAP 允许你根据需要使用 ACI(一般都称为 ACL 或访问控制列表)控制对数据读和写的权限,在安全方面比单纯的文件系统更有优势。综合 LDAP 的夸平台、容量、性能和安全考虑,我们的元数据存储平台使用 LDAP 而不是传统的文件系统。

对象分配、管理模块与元数据到 LDAP 信息映射模块和 LDAP 信息到元数据映射模块通过 get(struct object *)和 put(struct object)函数交互信息,保证了文件请求预处理/元数据管理模块和 LDAP/DBMS 自定义映射机制模块间的相对独立,前者不涉及任何 LDAP 信息的处理(strcut object 结构包括对象 id、顺序号、对应设备的 IP 地址、对象长度、对应的文件名、对应客户 id、对应客户组 id、创建时间、修改时间、更新时间等信息)。

3 性能分析及优化

尽管理论上轻量级目录访问协议(LDAP)具有更高的性能,但在实际应用当中由于受到实际系统的影响,在元数据读写量不大的情况下其性能反而不如 ext2 文件系统。这也是由于 ext2 使用了 Buffer Cache,在 TB 级环境中对文件的读写有优化。针对当前的应用环境,我们在元数据服务器中加入了自己的缓冲区,使性能得到了提升。测试条

(下转第 135 页)

生联系;文献[8]试图通过扩展一个面向对象的系统来对角色继承进行研究。然而,他们提出的角色继承的概念和继承机制都是基于对象技术,其本质只是用系统中扮演着各种角色的 Agent 来代替了原来的对象。正如我们在 2.3 节中分析的,多 Agent 系统中继承具有了新的特点,表达方式、分析过程都不同于对象技术中的继承。因此,在多 Agent 系统的分析和设计中,需要从 Agent 的特点和角色的概念出发研究继承的概念和继承机制,通过对继承的识别、描述和分析,在需求分析阶段得到清晰自然的系统结构层次图,促进设计阶段的软件重用。

但是,现有的研究还不足以使继承机制像对象技术中那样有效地支持软件系统的开发,还有一系列的问题需要解决。比如,本文仅对单继承进行了分析,实际上多继承也普遍存在于多 Agent 系统中;另外,本文讨论了怎样在需求分析阶段对继承进行分析和建模。下一步应该讨论在设计阶段对继承机制提供支持,甚至在面向 Agent 的程序设计语言中实现继承。

参考文献:

[1] Mao Xinjun, Yu E. Organizational and Social Concepts in Agent Oriented Software Engineering[A]. Proc of Agent Oriented Software Engineering[C]. 2004. 1-15.

[2] Ferber J, Gutknecht O, Michel F. From Agents to Organizational View of Multi-Agent Systems[A]. Proc of the 4th Int'l Workshop on Agent Oriented Software Engineering IV[C]. 2003. 214-230.

[3] Zambonelli F, Jennings N R, Wooldridge M. Developing Multiagent Systems: The Gaia Methodology[J]. ACM Trans on Software Engineering Methodology, 2003, 12(3): 317-370.

[4] Deloach S A, Wood M F, Sparkman C H. Multiagents Systems Engineering[J]. International Journal of Software Engineering and Knowledge Engineering, 2001, 11(3): 231-258.

[5] Bresciani P, Giorgini P, Giunchiglia F, et al. TROPOS: An Agent-Oriented Software Development Methodology[J]. Journal of Autonomous Agents and Multi-Agent Systems, 2004, 8(3): 203-236.

[6] Ferber J, Gutknecht O. A Meta-Model for the Analysis and Design of Organizations in Multi-Agent Systems[A]. Proc of 3rd Int'l Conf on Multi-Agent Systems[C]. 1998. 128-135.

[7] Depke R, Heckel R, Kuster J M. Improving the Agent-Oriented Modeling Process by Role[A]. Proc of the 5th Int'l conf on Autonomous Agents[C]. 2001. 640-647.

[8] Gottlob G, Schrefl M, Rock B. Extending Object-Oriented Systems with Roles[J]. ACM Trans on Information Systems, 1996, 14(3): 268-296.

(上接第 118 页)

件为 8K 个目录(记录),分为 2K 组,每组深度为 4 层,1K 个线程并发访问。

测试结果如图 2～图 5 所示。可以看出,未加缓冲区时,对于 ext2:第一次 10s 左右,第二次 5s 左右,后面的数据稳定在 0.2s 左右。对于 LDAP:第一次 10s 左右,第二次 5s 左右,后面的数据稳定在 2s 左右。加上缓冲区后 LDAP 性能优于 ext2。对于 ext2 来说,自制的缓冲区效果并不明显,这是因为 ext2 的 Buffer Cache 已经十分完善。而对于 LDAP 来说性能提高很明显。

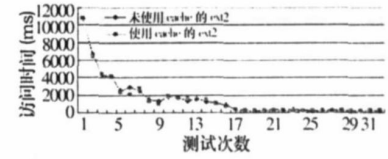

图 2 ext2 使用与未使用 Cache 的比较

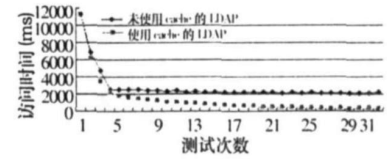

图 3 LDAP 使用与未使用 Cache 的比较

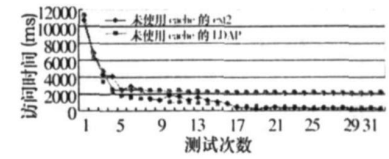

图 4 ext2 与 LDAP 未使用 Cache 的比较

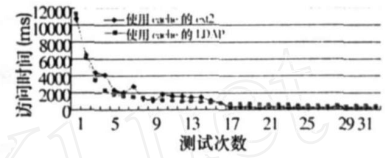

图 5 ext2 与 LDAP 使用 Cache 的比较

4 结束语

本文介绍了面向对象系统中元数据服务器的设计及元数据的组织和管理。该系统中元数据服务器使用了轻量级目录访问协议(LDAP)作为存放元数据的平台,针对这个平台设计了相应的数据分配算法和数据转换模块,并使用缓冲技术优化了性能。在容量、性能和安全等方面都得到了较好的效果。

在今后工作中,我们将在实测数据的基础上进一步改进系统,提高性能。

参考文献:

[1] Morris J H, Satyanarayanan M, Conner M H, et al. Andrew: A Distributed Personal Computing Environment[J]. Communications of the ACM, 1986, 29(3): 184-201.

[2] McKusick M K, Joy W N, Leffler S J, et al. A Fast File System for UNIX[J]. ACM Trans on Computer Systems, 1984, 2(3): 181-197.

[3] Gibson G A, Meter R V. Network Attached Storage Architecture[J]. Communications of the ACM, 2000, 43(11): 37-45.

[4] Wang Feng, Brandt S A, Miller E L, et al. OBFS: A File System for Object-Based Storage Devices[A]. Proc of the 21st IEEE / 12th NASA Goddard Conf on Mass Storage Systems and Technologies[C]. 2004.