

基于对象存储系统的服务质量框架研究*

冯 丹, 张 玲, 覃灵军

(华中科技大学 信息存储系统教育部重点实验室, 湖北武汉 430074)

摘 要: 简要介绍了基于对象存储系统的框架结构,并分析了系统的特点。在结合网络 QoS和存储 QoS的基础上,讨论了整个系统的 QoS要求。针对这些要求,提出基于对象存储系统的 QoS框架,分别在对象属性、OSD命令和 OSD设备上得以实现。

关键词: QoS 基于对象存储; 属性

中图法分类号: TP333

文献标识码: A

文章编号: 1001-3695(2007)01-0080-03

Study of QoS Framework for OBS System

FENG Dan ZHANG Ling QN Ling-jun

(Key Laboratory of Data Storage System for Ministry of Education, Huazhong University of Science & Technology, Wuhan Hubei 430074 China)

Abstract Architecture of Object-Based Storage System (OBSS) is briefly introduced and its characteristics are analyzed. Based on combination of network QoS and storage QoS, OBSS's QoS requirement is studied, and then OBSS's QoS framework is proposed, which is realized in attributes, OSD commands and Object-based Storage Device respectively.

Key words QoS; Object-Based Storage (OBS); Attribute

1 引言

基于对象的存储系统(以下称 OBS系统)已成为下一代互联网信息存储的核心技术,这种提供对象接口的方案完美地结合了 SAN 和 NAS 的优势,在系统性能、可扩展性、安全性、跨平台能力等方面都有了较大的提升。存储网络工业协会(SNIA)^[1]已于 2004 年提出了基于对象存储协议(OSD 协议)第九修订版,协议规定:对象是可变长的,对象中包含属性,用来描述对象的特征。对象的属性可以分为两种:①协议已经规定的,如 1h 为用户对象信息,3h 为用户对象的时间戳;②协议为用户和其他标准的定义所预留的,属性页号 80h~7FFFh 为用户定义的,8000h~EFFFh 可以给其他标准定义。因此,可以利用这些自定义的属性来实现 QoS。

系统所要求的 QoS 体现在从客户端到终端的过程中。例如,对于分组交换网的 QoS 问题,人们利用基于业务分类、定义优先级、资源预留、加权公平排队等策略的 Diff-Serve-MPLS 等现有技术来实现^[2]。对于终端磁盘的 QoS 问题,人们采用磁盘调度策略来加以保障。而在基于对象的存储系统中,有必要将网络的 QoS 和存储的 QoS 结合起来考虑,根据 OSD 协议和 SCSI 协议规范,有针对性地定义对象的属性,使 QoS 贯穿在建立连接、传送数据、存储数据的整个过程中。

2 OBS系统结构及特点

如图 1 所示, OBS 系统主要由四大部分组成,即客户端、元数据服务器、智能磁盘设备(OSD)和互联网络。客户端负责

向元数据服务器发起应用请求,如文件的创建、删除、打开、关闭、读写等;服务器在对客户进行身份确认后,将文件的元数据信息返回给客户端,包括文件的对象号、设备号、操作码等;然后客户端用得到的证书向 OSD 设备发送命令,建立连接;OSD 设备在对证书的完整性和正确性验证之后,就可以与客户端进行数据传输^[3]。

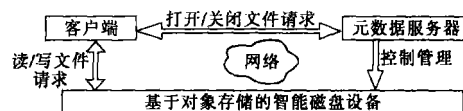


图 1 OBS 体系结构

OBS 系统的特点之一是提供了对象接口,对象由数据、属性和元数据构成。由于对象属性的加入,使得 OSD 可以掌握丰富的对象信息,从而实现优化高效的数据组织、自主的对象管理。最重要的是,智能存储设备还可以通过对象属性了解外部环境,合理地分配和预留,显著提高服务的质量^[4]。

3 OBS系统的 QoS 要求

与传统的基于块的存储系统不同,对象接口提供了足够丰富的属性来描述数据的特征,因此传统的提供最大努力的服务机制已经不能满足 OBS 系统的应用要求。OSD 协议可以根据具体的应用来提供不同的等级、不同种类的 QoS。此外, OBS 系统也将面临着更加复杂的外部环境,如一个 OSD 设备是与多个客户端互连的,每个客户端的请求都有不同的 QoS 要求,同一个客户端的请求也有不同的 QoS 要求,甚至同一个对象相对应不同的操作也会有不同的 QoS 要求,这些都需要智能 OSD 设备能根据对象的 QoS 要求,实时监测系统的 I/O 条件和负载能力,采取相应的准入策略^[5]。

结合 OBS 系统的特点,总结出三类 QoS 需求:

收稿日期: 2005-08-09 修返日期: 2005-11-29

基金项目: 国家“973”计划资助项目(2004CB318201)

(1)对象操作的 QoS。针对整个对象的操作,在这里定义的 QoS 属性为静态属性,即从对象的创建到对象的传输、存储、定义的属性都将伴随着这个对象,直到最后对象被删除才消失。例如,一个流媒体的文件对象,它从客户端传输到 OSD 设备需要很高的带宽,很低的抖动以及较低的时延,这些 QoS 要求在对象创建时就要建立,并且当 OBS系统的环境满足这些属性定义时,对象开始传送、存储。

(2)数据操作的 QoS 主要保证对象中数据的操作。这些 QoS 属性是动态属性,只有当某些应用需要对对象的数据进行操作时,才暂时在 OSD 命令中定义。例如,文件对象已经储存在 OSD 设备上,如果要作文件对象拷贝时,需要一个高带宽的磁盘 I/O 通道,就需要动态定义磁盘的带宽要求。再例如,在作增量备份时,客户端需要修改文件对象的一部分数据,修改过程的网络带宽要求、响应时间等也是动态定义的属性要求。

(3)属性操作的 QoS。对象中包括数据和属性,这类的 QoS 针对对象属性的操作,同样也是动态属性。例如在数据库的应用中,需要在规定时间内检索出符合一定要求(如修改时间)的对象 ID,并修改成客户定义的值,这个操作的响应时间就是需要动态定义的属性要求。

一般来说,一个对象的属性应始终包含对象操作的 QoS 但是不同的对象自定义的 QoS 会根据不同的应用而有所区别,如流媒体需要确保带宽,减少抖动,减少延时;商务应用应该保证响应时间以及数据的安全性和完整性,而数据操作的 QoS 和属性操作的 QoS 是在对这个对象具体操作时所定义的,有时可能会与对象操作的 QoS 冲突,此时应该以数据操作的 QoS 或属性操作的 QoS 为准。

4 OBS系统的 QoS框架

上面分析了 OBS系统的 QoS 要求,然后将基于这些要求提出 OBS 系统的 QoS 框架。

4.1 QoS的属性定义

OBS系统的一个特点就是对象属性可以自定义,因此可以用这些自定义的属性来丰富文件的特性,当然其中也包括整个对象的 QoS 需求。但是由于具体的应用所要求的 QoS 不尽相同,各个文件的属性定义也不一样,现以比较常见的流媒体为例子来说明对象 QoS 属性的结构。

首先确定流媒体文件的 QoS 属性需求: ①平均带宽。用来确保整个对象的传输速度,平均带宽可以通过公式计算出, $\text{平均带宽} = \frac{\text{文件对象大小}}{\text{平均传输时间}}$ 。②峰值带宽。由于流媒体的传输是通过传送相邻帧之间的差别而将视频压缩,根据图片细节的不同,所要传送的数据量随着时间的变化而变化。有时峰值带宽会远远大于平均带宽^[6],峰值带宽的确定可以使应用程序留出足够的缓冲区来满足 QoS 需求。③抖动。分组的时延变化用来确保流媒体的播放质量,如抖动太剧烈将使画面不平滑,甚至出现停顿的现象。

根据 OSD 协议,我们定义流媒体的 QoS 属性页如表 1 所示,属性页的格式如表 2 所示。

在 OSD 协议中规定,属性页号从 8000h~ EFFFh 是给其他标准保留的,现定义 8000h 为 QoS 属性页码号,同时也希望这

个定义能被 OSD 协议所规范。

表 1 QoS属性页

Attribute Number	Length (Bytes)	Attribute	Application Client Settable	OSD Logical Unit Provided
0h	40	Page Identification	NO	YES
1h	8	Average Bandwidth	YES	NO
2h	8	Peak Bandwidth	YES	NO
3h	8	Jitter	YES	NO
4h~ FFFFFFFh		Reserved	NO	

表 2 QoS属性页格式

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)	Page Number(2h)						(LSB)
3								
4	(MSB)	Page Length(18h)						(LSB)
7								
8	(MSB)	Average Bandwidth						(LSB)
15								
16	(MSB)	Peak Bandwidth						(LSB)
23								
24	(MSB)	Jitter						(LSB)
32								

4.2 OSD命令

上面分析了 OBS系统的 QoS 要求,具体的每个文件对象都会有其固定的 QoS 属性,即静态属性。同时,对这个对象具体操作时,如对数据操作或对属性操作,同样地要对这些操作定义 QoS 属性,这些属性为动态属性。当对象既有静态属性又有动态属性时,需要通过某些机制来保证实现哪个等级的 QoS。

客户端发送 CDB (Command and Description Block 命令表述块)给 OSD 设备,可以通过扩展 CDB 命令第 11 个字节的定义来实现静态属性和动态属性的区别。具体定义如表 3 所示。11 个字节的第 4 位和第 5 位的值为 01b 时,表示后面的属性为 QoS 动态属性。

表 3 Get and Set Attributes CDB format Code Values

Value	Description
00b	Reserved
01b	Attributes Value is Specified Temporarily
10b	Get an Attributes Pages and Set an Attribute Value
11b	Get and Set Attributes Using Lists

动态属性应该有效时限,OBS系统只是在这个期间内保证这些 QoS 的动态属性,为此定义一个标志位来表示这个期间的开始和结束。同样利用 CDB 命令的第 11 个字节来定义第 6 位、第 7 位的值,当值为 11b 时,表示动态属性有效时限的开始;值为 00b 时,表示有效时限的结束;值为 01b 和 10b 保留。

只有当 11 个字节的第 4 位和第 5 位值为 01b 时,第 6 位和第 7 位的值才有意义。

4.3 OSD设备端的 QoS策略

OSD 是一个智能设备,包括处理器、RAM 内存、网络接口、存储介质如磁盘等,OSD 可以根据处理器的计算能力来制定优化策略,保证设备端的 QoS 性能。在这里,简单讨论一下准入策略和资源预留策略。

4.3.1 准入策略

由于多个客户端有可能同时连接一个 OSD 设备要求读写对象,那么每个客户的 QoS 不是都能保证的,OSD 设备采用准入策略。当一个新的客户请求到来时,OSD 设备先判断网络条件和磁盘条件是否满足,在应答信息中告知客户端。准入策

略有三类^[7]:

(1)最大努力策略。努力满足所有客户的请求,当 OSD 设备性能达到极限时,整个系统的 QoS 恶化,最大努力策略将首先降低不重要应用的 QoS。

(2)确定策略。OSD 设备根据客户请求的 QoS 判断接受哪些请求,拒绝哪些请求。这种方式一般都假设最坏情况,它可能会导致较低的网络和磁盘利用率。例如,对流媒体对象,如果根据 QoS 属性的峰值带宽来确定是否准入,那么肯定会有比例网络带宽的浪费。确定策略一般是 QoS 的硬件保证。

(3)统计策略。OSD 设备实时检测系统的性能,用软件的方式来实现 QoS^[8]。系统可以结合确定策略和统计策略来提高系统的利用率,用确定策略来保证一些重要应用的 QoS 用统计策略来实现其他应用的 QoS。

4.3.2 资源预留策略

因为对象的 QoS 性质由属性定义, OBS 系统可以提前根据这些属性来分配网络资源和磁盘的 I/O 资源。资源预留策略只是为不同的 QoS 需求分配不同比例的传输带宽。例如,当多个客户端要同时对同一个对象文件进行读写操作(即热点数据),系统就需要为这个热点数据预留较多的磁盘 I/O 带宽,并且当预留的资源没有被充分利用时,系统应该可以回收这些多余的资源,进行再分配,资源预留过程是动态的。对整个系统来说,负载比较少时,资源预留策略能较好地提供 QoS,负载增加,它提升性能的能力急剧下降。提升系统 QoS 性能的幅度与系统负载成反比。

5 结束语

随着技术的发展,用户对应用的 QoS 要求将会越来越高。OBS 系统的 QoS 要求可分为三个方面,即对象的 QoS 操作、数据的 QoS 操作和属性的 QoS 操作,三者各有联系,也各有区

别。作为下一代互联网信息存储的核心技术, OBS 系统在保证应用的 QoS 方面体现出更多的优势。属性的灵活定义使用户应用的 QoS 范围得到大大的扩展, OSD 命令的扩展确定了对象静态属性和动态属性的作用期限,而 OSD 设备作为智能设备也提供了高于一般磁盘设备的 QoS 性能保证。

参考文献:

- [1] Object Based Storage Devices Command Set (OSD) [EB/OL]. <http://www.t10.org/drafts.htm> T10 Working draft 2004-01.
- [2] 赵慧玲, 徐向辉. QoS 的下一代网络结构框架的研究 [EB/OL]. <http://www.chinatelecom.com.cn>, 2003-02.
- [3] Mesnier G R, Ganger E R. Object-based Storage [J]. IEEE Communications Magazine, 2003, 41(8): 84-90.
- [4] D Elkar, et al. Passive NFS Tracing of an E-mail and Research Workload [C]. San Francisco Conf File and Storage Tech., 2003: 203-205.
- [5] Yingping Lu, David H C Du. Tm Tuart QoS Provisioning Framework for an OSD-based Storage System [C]. Proceedings of the MST, 2005.
- [6] Larry L Peterson, Bruce S Davis. Computer Networks a Systems Approach (2nd edition) [M]. China Machine Press, 2000: 18-23.
- [7] Z Dimitrijevic, R Rangaswami. Quality of Service Support for Realtime Storage Systems [C]. Stefan Proc. of Intl. PSI Conference, 2003.
- [8] Z Dimitrijevic, R Rangaswami, E Chang. The XTREAM Multimedia System [C]. Proceedings of the IEEE Conference on Multimedia and Expo., 2002: 545-548.

作者简介:

冯丹,女,博导,主要研究方向为计算机系统结构、磁盘阵列技术、海量存储、网络存储;张玲,男,硕士研究生,主要研究方向为网络存储系统、磁盘驱动;覃灵军,男,博士研究生,主要研究方向为网络存储、虚拟存储、海量存储技术。

(上接第 79 页) 访问构件、数据库和基类窗体设计好之后,将开发人员分为两个小组,每个小组负责三个线索,由于各线索之间没有相互调用,小组之间既不会相互影响,还可以实现相似功能的代码重用,大大提高了编程效率,缩短了开发周期,节约了开发成本。

在系统的演化过程中,新增线索时不会对其他部分产生不良影响,还可以重用其他共享类和构件,充分显示了正交软件体系结构的优越性。

该系统在服务站内部采用 C/S 模式,在与服务部、配件供应商之间进行数据交换时采用 B/S 模式,适合中小型汽车生产厂家售后服务采用第三方软件的实际需求。目前该系统已经在某汽车生产厂的十多个售后服务站应用,并且应用规模正不断扩大。

3 结论

本文依据正交软件体系结构思想,叙述了正交软件体系结构的设计和演化控制方法,结合正交软件体系结构的应用实例,论证了正交软件体系结构的实践可行性及其优越性,是一种实用的软件体系结构设计模型。在实际应用中,对正交软件体系结构的设计根据需求因地制宜,在不能采用完全正交时,

适当变通,不过分追求完全正交,只要满足高内聚,低耦合即可。事实证明,将正交软件体系结构的线索按层次划分为多级,有利于功能划分和粒度控制,使软件结构更加清晰,使开发工作量分配均匀,做到了开发工作的高效有序、易修改、可维护性强,能方便地实现结构调整;便于移植,重用粒度大,可以实现体系结构级的重用。

参考文献:

- [1] 左爱群,黄水松.软件体系结构的发展与研究[J].计算机工程与应用,1999,35(3):19-21.
- [2] 张友生.软件体系结构[M].北京:清华大学出版社,2004:72-76.
- [3] 谭凯,林子禹,彭德纯,等.多级正交软件体系结构及其应用[J].小型微型计算机系统,2000,21(2):138-141.
- [4] 张友生,陈松乔.正交软件体系结构的设计与演化[J].小型微型计算机系统,2004,25(2):295-299.
- [5] 周莹新,艾波.软件体系结构建模研究[J].软件学报,1998,9(11):866-872.

作者简介:

李海洋(1972-),男,讲师,硕士研究生,研究方向为网络化制造;李柏林(1962-),男,教授,博导,主要研究方向为网络化制造;郭荣佐(1973-),男,硕士,主要研究方向为嵌入式系统、软件系统理论、计算机连锁。