VISA 存储网络系统元数据管理的研究

冯 丹 叶 飞 施 展

(华中科技大学 计算机科学与技术学院, 湖北 武汉 430074)

摘要: 为了提高虚拟接口存储结构(VISA)存储网络系统的性能,设计实现了基于块级的元数据管理子系统. 该系统的元数据管理子系统实现了物理设备和逻辑设备的块地址空间之间的映射,支持动态地址映射、动态块重分配和热块冗余技术,可以对 I/O 请求的路径和数据传输的路径完全控制,从而有效地改善了 VISA 存储网络系统的 I/O 性能,使数据布局更加合理,提高了存储管理的灵活性,初步实现了负载均衡,并且使VISA 存储网络系统具备了容错能力和节点的动态增加能力. 测试结果表明:在加入了基于块级的元数据管理功能带来相应开销的情况下,VISA 存储网络系统原型与传统存储网络系统 iSCSI + Linux MD 相比,传输延迟减少了 8.2 %～21.0 %,传输吞吐率提高了 8.7 %～11.9 %.

关 键 词: 虚拟接口存储结构; 网络存储系统; 元数据管理

中图分类号: TP333 **文献标识码**: A **文章编号**: 1671-4512(2007)03-0049-04

Metadata management of VISA storage network systems

Feng Dan Ye Fei Shi Zhan

(College of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China)

Abstract: To enhance the performance of VISA (virtual interface based storage architecture) storage network system, a new metadata management subsystem is proposed. It is running at block-level. The high efficiency and scalability come from the following mechanisms: a clear-hierarchy mapping from physical to logical blocks, dynamic address mapping, dynamic block reallocation and hot block redundancy. Furthermore, the proposed subsystem can perform a full control over the data request and transfer path. VISA storage network system is improved with a more reasonable data layout, the flexibility of storage management and preliminary load balancing. It is also brought the functionality of fault-tolerant and the ability to add storage resource dynamically. The experiment result proves that, even though the proposed subsystem may bring more overhead, VISA still outperforms traditional storage network system such as iSCSI with Linux MD in both transfer delay and throughput, the increase is 8.2 % to 21.0 % and 8.7 % to 11.9 % respectively.

Key words: virtual interface storage architecture; network storage system; metadata management

网络数字信息爆炸性的增长产生了海量的数据信息,频繁地存取和使用这些数据成为提高系统性能的瓶颈,其中以对元数据的存取访问为甚. 虽然元数据相对于整个存储系统的容量来说不大,但是所有文件系统中 50 %～80 %的存取访问是针对元数据的,因此有效地管理元数据是很重要的.

传统上,元数据和数据由同一文件系统管理,存储在同一台机器或同一个设备上. 为了提高效率,元数据经常保存在物理上离它所描述的数据

收稿日期: 2005-10-27.

作者简介: 冯 丹(1970-),女,教授;武汉,华中科技大学计算机科学与技术学院 (430074).

E-mail: dfeng@mail.hust.edu.cn

基金项目: 国家重点基础研究发展计划资助项目 (2004CB318201); 国家自然科学基金资助项目 (60303032).

很近的地方.在一些现代的分布式文件系统中,数据存储在可以通过网络直接存取的设备上,元数据则由一个或多个元数据服务器单独管理[1].

元数据的管理可以是基于块级、文件级和对象级的.在典型的基于块级的元数据管理中,数据块被映射到一个或多个存储设备上,块地址可能会分布在多个存储设备上,但呈现给用户的是一个单独的逻辑设备或逻辑卷[2].

本文研究虚拟接口存储结构(VISA)存储网络系统元数据管理的灵活性和性能,其灵活性通过元数据服务器的动态地址映射来证明,其性能通过 ping-pang 通信测试程序对原型系统和 iSCSI+Linux MD 分别进行测试评估.阐述了VISA存储网络系统的元数据管理部分的设计及实现策略,测试结果表明使用基于块级的元数据管理,可以在较小的性能开销下,实现系统功能层次上的提升.

1 VISA 存储网络系统的体系结构

体系结构包括三个部分:一台或者多台元数据服务器(在原型系统中,只有一台元数据服务器)、客户端节点和存储节点,图1描述了 VISA 存储网络系统的体系结构.

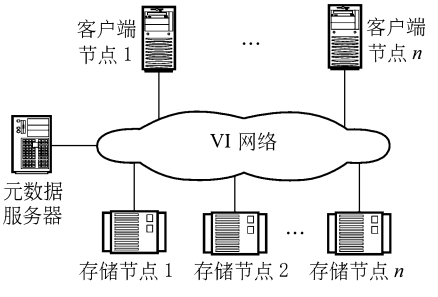

图1 VISA 存储网络系统结构

在 VISA 存储网络系统的原型中,元数据服务器和存储节点是以 Linux 内核模块的形式实现的,客户端节点是以 Linux 块设备驱动的形式实现的,底层通信协议采用的是内核级虚拟接口结构(VIA)通信协议[3].原型实现了 RAID-0 级地址映射语义和热块冗余技术,客户端节点存取大块数据的时候可以并行访问多个存储节点;还实现了动态块重分配技术,支持动态扩容功能.

2 元数据管理子系统的设计

基于块级的元数据管理可以实现物理设备和逻辑设备的块地址空间之间的层次清楚的任意映射;对 I/O 请求的路径和数据传输的路径完全控制,实现负载均衡;可以支持动态块重分配、故障恢复和动态扩容,使数据布局更为合理.

2.1 动态地址映射

存储节点上每个连续的块地址空间称为一个存储实体(SE).每个存储实体对应一张块位图(PBB),以反映存储资源的使用情况.

VISA 存储网络系统通过逻辑块地址到物理块地址转换的地址映射表(AMT)来支持动态地址映射.块设备地址空间的转换根据需要可以采用不同的映射策略.块位图所表示的块地址空间和存储实体上的实际物理块地址空间一一对应,而地址映射表则是经过转换的逻辑块地址空间.

假设局域网里面有 n 个网络节点,不同的网络节点的媒体访问控制(MAC)地址 A 是不同的,用元组 $B_{SE}(A, I, O)$ 表示存储实体上的块,其中 I 表示存储实体的 ID,O 表示实体内偏移地址,用 $B_{AMT}(A, I, O, O')$ 表示地址映射表 AMT 中的块,O 表示映射后的实体内偏移地址,则动态地址映射策略可以用函数 F 表示为

$$O' = F(A_i, I_{ik}, O_{ik}), \quad (1)$$

式中:A_i 表示第 i 个网络节点的 MAC 地址,$i \in [0, n-1]$;I_{ik} 表示 MAC 地址为 A_i 的网络节点上的第 k 个存储实体的编号,同一网络节点上不同存储实体的编号是不同的;O_{ik} 表示存储实体 I_{ik} 的块地址空间的偏移,假设存储实体的块大小为 B byte,块数为 C,则 $O \in [0, C-1]$;O 即物理块 $B_{SE}(A_i, I_{ik}, O_{ik})$ 被映射到 AMT 后的块偏移地址.

把由 O 组成的连续块地址空间称为物理地址空间,把由 O 组成的连续块地址空间称为逻辑块设备的可用地址空间.式(1)表明,元数据服务器是根据某种策略对地址空间进行转换,而不是将 I/O 路径从一个设备定位到另一个设备.

若把同一个存储节点上的不同存储实体映射到同一个地址映射表,则当客户端要存取大量数据时,可能会使大量的 I/O 请求集中在同一存储节点上,而其他存储节点可能处于负载比较轻的状态,这样就不能有效利用网络带宽.为了避免这种情况,根据式(1),可以在地址映射策略的实现中进行检测,若对于同一 MAC 地址 A 出现不同的 I,则向用户提示,尽量避免这种情况出现.

2.2 动态块重分配

在独立磁盘冗余阵列(RAID)的故障恢复模型中,若少于或等于 D 个磁盘,则出现故障还能够继续正常工作.若多于 D 个磁盘出现故障,则

RAID 完全不可用. 在大多数 RAID 系统中, D 值通常为 1. VISA 存储网络系统在地址映射策略中引入了 RAID 映射语义. 当多于 D 个存储节点出现故障时,该地址映射表所对应的逻辑设备将不能够正常工作,但是不影响系统中的其他逻辑设备. 在某个存储实体失效时或给某个逻辑设备增加块地址空间时,意味着某个逻辑设备的块地址空间会发生变化,这时就需要采用动态块地址重映射[4,5].

在下文中实现了 VISA 存储网络系统的动态块重分配功能,并以 RAID-1 级映射策略为例. 图 2 和图 3 描述了系统按照 RAID-1 级语义进行故障恢复的过程. 存储实体 SE2 发生故障后,虽然不能对 SE2 访问,但系统数据仍然是完整的. 块 1-1、块 1-2、块 1-3 和块 2-1′、块 2-2′、块 2-3′成了孤本,在元数据服务器检测到 SE2 发生故障后,重新映射地址映射表,并为成为孤本的块创建副本,从而使系统仍然维持 RAID-1 级语义.

SE1			SE2			SE3	
1-1	3-1′		2-1	1-1′		3-1	2-1′
1-2	3-2′		2-2	1-2′		3-2	2-2′
1-3	3-3′		2-3	1-3′		3-3	2-3′

(a) 存储实体 1　(b) 存储实体 2　(c) 存储实体 3

图 2　存储系统正常情况下的数据块布局

SE1			SE3	
1-1	3-1′		3-1	2-1′
1-2	3-2′		3-2	2-2′
1-3	3-3′		3-3	2-3′
2-1″	2-2″		1-1′	1-2′
2-3″			1-3′	

(a) 存储实体 1　(b) 存储实体 3

图 3　存储系统故障恢复之后的数据块布局

逻辑设备的地址空间越大,对地址映射表的平均查找时间就越长,所以不应一次性为逻辑设备分配过多的存储资源. 而动态扩容功能可以根据需要先分配初始大小的存储资源给逻辑设备,当空闲空间低于一定的比例时,再为该设备分配更多的存储资源.

通过遍历地址映射表 AMT 中的每一个存储节点检测其状态,若检测到存储节点发生故障,则进行恢复.

2.3　热块冗余技术

数据放置不当可能会对系统性能带来负面影响. 研究表明:跨越多个磁盘的数据分块要优于复杂的文件放置算法[6]. 如果把用户数据分布在系统的多个存储节点上,那么系统性能肯定会得到提高,但同时会牺牲较多的存储空间. 因此就产生数据放置策略的问题:哪些数据需要复制,用户数据的副本放在哪里,什么时候创建这些副本.

由于存取访问的局部性而且 90 %的 I/O 请求集中在 10 %的数据块上[7],考虑到对某些数据块的访问比较频繁(即热块),而对某些块的访问频率相对较低(即冷块),如果把热块数据只存放在单个存储节点上,那么热块数据所在的存储节点负载比较重时,该存储节点的响应时间就会增加,导致整个系统性能下降而成为瓶颈. 因此,采用热块冗余技术对热块数据进行冗余备份,可有效地提高系统的性能和可靠性.

VISA 存储网络系统将存储实体性能优异的部分作为热区,用来保存热块数据及其副本.

客户端节点在处理 I/O 请求时,对相关的块进行计数,超过某个阈值时就把该块作为热块、并标记,在该块被更新时或者发生写操作时,将该块及其副本写到与该逻辑设备相关联的存储实体的热区[8]. 图 4 描述了热块冗余备份的过程.

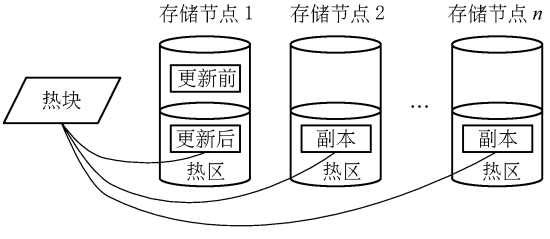

图 4　热块冗余备份过程

热块冗余备份主要是优化 VISA 存储网络系统的读性能,故在此分析其读操作. 假设在未进行热块备份之前,热块(不计冗余部分)占 VISA 存储网络系统中总的数据块数的百分比为 u,存储节点对热区的读操作平均时间为 T_{hot},对冷区的读操作平均时间为 T_{cold},优化后单存储节点上读操作的性能加速比

$$S_{READ} = T_{cold}/(uT_{hot} + (1 - u)T_{cold}).$$

假设 VISA 存储网络系统中有 n 个存储节点,网络带宽为 M,存储节点分块大小为 B,客户节点和存储节点收发读命令的开销均不计且硬件配置相同. 客户节点到存储节点的平均网络传输时间为 T_t,存储节点读取 B 的数据的平均时间为 T_{sm},网卡接收和发送的时间均为 B/M,客户节点收到数据后处理的开销为 T_c,则客户节点读 B 的数据的平均时间

$$T_{cm} = 2T_t + T_{sm} + 2B/M + T_c.$$

假设对 n 个存储节点上的备份进行并行访问,客户节点从每个存储节点读取 B 的数据,在

理想状况下的平均时间 T_{pcm} 及加速比 S_p 分别为:

$$T_{pcm} = 2T_t + T_{sm} + (1 + n)B/M + nT_c,$$

$$S_p = nT_{cm}/T_{pcm} = n(2T_t + T_{sm} + 2B/M + T_c)/(2T_t + T_{sm} + (1 + n)B/M + nT_c). \quad (2)$$

根据式(2),在理想状态下,n 个存储节点上的读数据时间相当于减少到原来的 $1/n$,同时,热区上的数据越多,平均读数据时间越短.

3 性能评估

测试平台配置如下:Intel Xeon(TM) 3.00 GHz CPU,1 Mbyte 缓存,512 Mbyte 内存,Barracuda 7200.7-ST380011A 80 Gbyte Ultra ATA/100 硬盘,2 Mbyte 缓存,SysKonnect SK-9821 千兆以太网卡,32 bit 33 MHz PCI 插槽;Red Hat Linux 7.3 操作系统,2.4.18-3-i686-up 内核.采用 Cisco Catalyst 3750 千兆交换机连接的星形 Gigabit 以太网.

本研究采用的测试程序是 ping-pang 通信性能测试程序,对 VISA 存储网络系统原型和 iSCSI+Linux MD 在不同数据包大小情况下的传输延迟和吞吐率分别进行了测试.VISA 存储网络系统原型采用仿 RAID0 级地址映射策略,配置两个存储节点、客户节点和元数据服务器节点各一个;iSCSI 环境的配置采用两台测试机器配 Intel v20 iSCSI target,一台测试机器配 Intel v20 iSCSI initiator和 Linux MD.

根据图5和图6的测试结果,相对于 iSCSI+Linux MD 存储网络系统,VISA 存储网络系统原型无论是在传输延迟还是在吞吐率的对比上都占有优势.VISA 存储网络系统的传输延迟减少了 8.2 %～21.0 %,传输吞吐率提高了 8.7 %～11.9 %.考虑到以太网对物理带宽的限制及元数据管理带来的开销,VISA 存储网络系统在传输性能方面可以说很好,而且基于块级的元数据管理功能是 iSCSI 等存储网络系统所没有的.

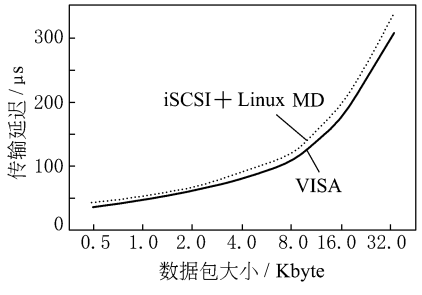

图5 不同数据包大小下传输延迟的对比

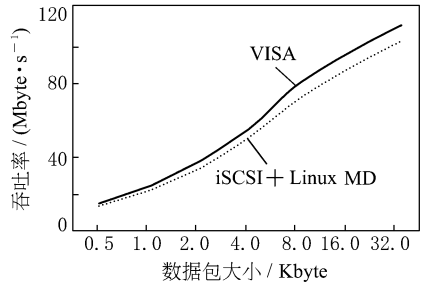

图6 不同数据包大小下的吞吐率的对比

参 考 文 献

[1] Brandt S, Xue L, Miller E, et al. Efficient metadata management in large distributed file systems[C]//20th IEEE/11th NASA Goddard Conference on Mass Storage Systems and Technologies. Los Alamitos: IEEE Comput Soc, 2003: 290-298.

[2] Flourish M D, Anastasiadis S V, Bilas A. Block-level virtualization: how far can we go?[C]//Proceedings of the Second IEEE-CS International Symposium on Local to Global Data Interoperability-Challenges and Technologies. Los Alamitos: IEEE Comput Soc, 2005: 98-102.

[3] 黄浩丹.分布式存储系统的访问接口设计[D].武汉:华中科技大学计算机科学与技术学院, 2004.

[4] Flourish M D, Bilas A. Violin: a framework for extensible block-level storage[C]//Proceedings of the 22nd IEEE/13th NASA Goddard Conference on Mass Storage Systems and Technologies. Los Alamitos: IEEE Comput Soc, 2005: 128-142.

[5] Flourish M D, Bilas A. Transparent data versioning at the block I/O level[C]//12th NASA Goddard, 21st IEEE Conference on Mass Storage Systems and Technologies. Los Alamitos: IEEE Comput Soc, 2004: 315-328.

[6] Ganger G R, Worthington B L, Hou R Y, et al. Disk subsystem load balancing: disk striping vs. conventional data placement[C]//Proceeding of the Twenty-Sixth Hawaii International Conference on System Sciences. Los Alamitos: IEEE Comput Soc, 1993: 40-49.

[7] Kazuhiko M, Masaru K. Hot block clustering for disk arrays with dynamic striping[C]//The 21st VLDB Conference. San Francisco: Morgan Kaufmann Publishing, 1995: 90-99.

[8] Weil S A, Pollack K T, Brandt S A, et al. Dynamic metadata management for petabyte-scale file systems[C]//Proceedings of the 2004 ACM/IEEE Conference on Supercomputing. Los Alamitos: IEEE Comput Soc, 2004: 4-14.