第 34 卷 第 6 期 计 算 机 工 程 2008 年 3 月
Vol.34 No.6 Computer Engineering March 2008
·软件技术与数据库· 文章编号：1000—3428(2008)06—0074—03 文献标识码：A 中图分类号：TP391

高性能逻辑文件系统设计与实现

赵 奕[1,2,3]，唐荣锋[1,2,3]，陈 欢[1,2,3]，熊 劲[1,2]，马 捷[1,2]
(1. 中国科学院计算技术研究所国家智能计算机研究开发中心，北京 100080；
2. 中国科学院计算技术研究所计算机系统结构重点实验室，北京 100080；3. 中国科学院研究生院，北京 100039)

摘 要：服务器端文件系统不仅需要很大的容量，而且要为大量并发访问提供很高的 I/O 性能。该文提出一种把多个物理文件系统通过软件集成为一个逻辑文件系统的技术，很好地聚合了各个文件系统所在磁盘设备的带宽和容量，综合了不同文件系统在元数据和数据处理性能上的优势。性能测试结果表明，逻辑文件系统技术是一种构造支持高度并发访问的高性能文件系统的有效方法。

关键词：文件系统；数据布局；元数据

Design and Implementation of High Performance Logical File System

ZHAO Yi[1,2,3], TANG Rong-feng[1,2,3], CHEN Huan[1,2,3], XIONG Jin[1,2], MA Jie[1,2]
(1. National Research Center of Intelligent Computing Systems, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080;
2. Key Laboratory of Computer System and Architecture, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080;
3. Graduate University of Chinese Academy of Sciences, Beijing 100039)

【Abstract】File systems on the servers require both massive capacity and high I/O performance for large quantities of concurrent accesses. In this paper, a technology of building a logical file system by integrating a number of physical file systems is presented. It integrates the bandwidth and capacity of those physical file systems, and combines the performance of both meta-data and file data from different kinds of existing file systems. Performance evaluation result shows that the logical file system technology is an effective way of building high performance file system with the ability to support highly concurrent accesses.

【Key words】file system; data layout; meta data

高性能的存储服务器和 Web 服务器需要高性能的文件系统作为支撑。服务器端文件系统不仅需要支持很大的容量，而且需要提供很高的 I/O 性能(包括元数据处理性能和读写带宽)，特别是对大量并发访问提供良好的支持。存储服务器一般都配备比较多的磁盘设备和磁盘控制器，以及大量的内存。如何充分利用这些硬件来实现一个支持高度并发访问的高性能文件系统，是一个值得研究的问题。现有的技术主要是通过把多块磁盘虚拟成一个块设备的方法来实现，用户可以在这个集合多个物理磁盘的虚拟块设备上，创建普通的文件系统。本文提出了实现高性能服务器端文件系统的另一种方法，即在现有文件系统的基础上，建立一个逻辑文件系统，它聚合多个普通的物理文件系统的性能和容量。逻辑文件系统给用户提供一个全局名字空间和标准的访问接口(系统调用)。在逻辑文件系统内部，通过文件的形式存储元数据和数据，使用可配置的数据分布策略来管理数据的存储。逻辑文件系统充分利用了多个物理文件系统的聚合性能，提供了一个高性能、高扩展性、高并发度的服务器端文件系统解决方案。

1 研究背景

为了解决单个磁盘设备的性能局限，大量的研究和产品实现了把多个物理磁盘设备集成、虚拟成一个单独的块设备的功能。根据其实现的方法不同，可以分成以 Raid[1]控制器为基础的硬件方案和以 LVM[2]为基础的软件方案。这些产品的共同特点是，把多个磁盘设备虚拟成一个块设备，用户可以在这个块设备上建立普通的文件系统，例如 ext3 或者是 xfs。在 LVM 设备驱动层或者硬件 Raid 卡上，对于虚拟块设备的读写访问被分散到多个物理磁盘设备上，从而加大了并发访问的吞吐率。这种方法存在的弱点包括：单个 Raid 控制器提供的带宽受到通道带宽的限制，单个文件系统在比较多的线程同时读写的情况下，并发访问性能较差。

另外一些研究工作集中在把多个文件系统整合为一个文件系统。文献[3]把分散在多个文件系统上的目录集合到一个目录下，便于数据的管理和使用，同时为一些特殊的应用场景，如对一个只读的介质进行虚拟的写操作等。与本文的逻辑文件系统相比，Unionfs 的主要目标是多个目录内容的使用和管理的便利性而不是集成性能，且 Unionfs 并不维护单一的目录视图，用户可以看见其集成的文件系统上的内容。

Raif[4]也是在多个现有的文件系统上，利用 Raid 的概念和技术，建立一个虚拟的文件系统，用以集成多个文件系统的存储和性能，并且可以在其上扩展出加密、压缩等高级特性。Raif 与本文的逻辑文件系统的主要区别在于元数据的处理上，Raif 在多个磁盘上分布存储文件的元数据，实现比较复杂，而且一致性较难维护，也难以发挥专用的元数据文件系统的性能优势。

现有技术还不能很好地解决服务器端高度并发文件访问的性能，为此本文提出一种聚合多个物理文件系统的性能和容量的逻辑文件系统技术。

基金项目：中国科学院创新基金资助项目“新一代机群关键技术的研究”(KGCX2-SW-116)
作者简介：赵 奕(1980－)，男，硕士研究生，主研方向：机群文件系统；唐荣锋、陈 欢，博士研究生；熊 劲、马 捷，博士、副研究员
收稿日期：2007-03-30 **E-mail：**zhaoyi@ncic.ac.cn

2 逻辑文件系统的设计目标

逻辑文件系统的设计目标包括以下几个方面:

(1)提供一个与 VFS 接口相兼容的内核级的文件系统,使用逻辑文件系统的应用程序不需要与特殊的库函数连接,也不需要重新编译。

(2)聚合多个物理文件系统的性能和容量。逻辑文件系统的聚合读写带宽和存储容量应与直接操作下层的物理文件系统的带宽和容量之和相当。

(3)支持高并发度访问,在大量的并发读写的情况下,保持较高的聚合带宽。

(4)良好的元数据性能。

(5)对于下层物理文件系统有良好的兼容性,不与特定的现有文件系统绑定。

(6)完整支持 NFS[5],作为 NFS 后端文件系统,提供良好的性能。

3 逻辑文件系统的整体结构

逻辑文件系统是一个软件抽象层,用来把多个文件系统集合成一个单独的逻辑文件系统。用户通过 VFS 访问逻辑文件系统的功能。

逻辑文件系统的结构如图 1 所示,它由多个物理文件系统构成,其中一个物理文件系统存储逻辑文件系统的元数据,称为元数据文件系统,其他物理文件系统用于存储逻辑文件系统的文件内容(数据),称为数据文件系统。

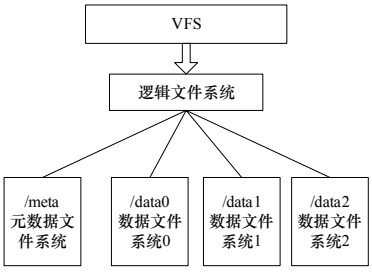

图 1 逻辑文件系统整体结构

逻辑文件系统的特点是其元数据和数据都是以文件的形式保存在下层的物理文件系统上。这样设计的优点在于:(1)逻辑文件系统可以不必处理复杂的磁盘布局管理,而专注于处理数据分布策略、IO 预测与调度、可靠性等更高层的任务;(2)可以利用现有的文件系统高度优化的元数据和数据操作,避免从头实现所需要的优化工作。

逻辑文件系统的另一个特点是把元数据和数据分开,分别用不同的下层文件系统来存储。相对于在每个下层文件系统上混和存储数据和元数据,优点在于:(1)方便元数据的访问,避免跨越多个磁盘维护元数据目录结构的复杂性;(2)利用不同的下层文件系统来存储数据和元数据,用元数据性能比较好的下层文件系统(如 reiserfs)来存储元数据,用读写性能比较好的下层文件系统来存储数据,充分挖掘性能优势。

4 逻辑文件系统的关键技术

4.1 元数据存储和处理

元数据是描述文件本身特性的数据,例如文件名、文件大小、文件的修改时间、文件所在的目录结构等。逻辑文件系统以附着属性的方式直接附着在下层的元数据文件系统的文件上。

文件系统有和逻辑文件系统完全一样的目录结构。每个逻辑文件系统的文件或目录的元数据都用一个元数据文件来存储,元文件是存储在元数据文件系统上的同名的文件或目录。元文件与逻辑系统文件有完全一样的各种属性,包括文件名、所在目录、索引节点号、文件时间、读写执行权限等。这样,对逻辑文件系统元数据的操作,最终转化为对元数据文件系统的元文件的操作,读取逻辑文件系统的元数据就是读取元数据文件系统的元数据。逻辑文件系统元数据存储系统如图 2 所示。

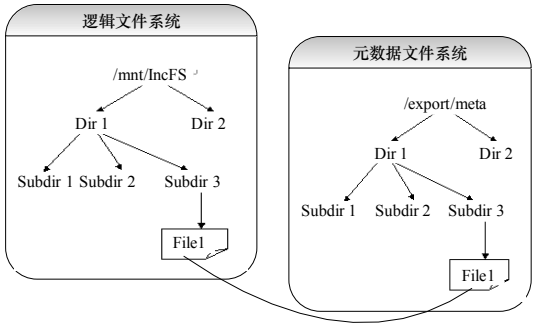

图 2 逻辑文件系统元数据存储

4.2 数据存储和布局

逻辑文件系统的数据是以文件的形式存放在多个下层的数据文件系统上,以逻辑文件系统文件的索引节点号码和布局编码等信息,作为数据文件系统上存储数据文件的文件名。当用户读写逻辑文件的数据时,逻辑文件系统就打开数据文件系统上对应的文件,读写数据文件里面的数据。

逻辑文件系统的文件数据,可以根据不同的需求,以不同的分布方法存储在数据文件系统上,称为数据的布局(layout)。考虑到没有任何一种分布策略能够完全适用于所有的应用场合,因此,在该逻辑文件系统中提供多种不同的布局策略。

最简单的布局是把单个逻辑文件的所有数据都存储在同一个数据文件系统上,但是不同逻辑文件的数据存储在不同的数据文件系统上,称为 Single 布局。Single 布局的主要优点是简单、高效。每一个逻辑文件只对应于一个数据文件,这样,所有对逻辑文件数据的读写操作都可以直接转发到对数据文件的读写操作。Single 布局的缺点是:如果只读写一个文件,则无法充分利用多个数据文件系统的存储容量和性能。

另一种数据布局为 Stripe。Stripe 布局把逻辑文件的数据划分成小块,然后把这些小块的数据分散存储在多个数据文件系统上。Stripe 模式的主要优点是:可以充分利用多个数据文件系统的容量和聚合带宽;缺点是:(1)比较复杂;(2)把数据分散到所有的数据文件系统上,如果其中的一个文件系统出现了故障,那么所有的文件都会被破坏。

为了尽可能均衡各个下层数据文件系统的访问,增加整体的并发度,逻辑文件系统采用了将不同的文件尽量分散存储在不同数据文件系统上的策略。在逻辑文件系统的文件创建的时刻,用一个全局的分配算法来决定文件的数据存储在哪些数据文件系统上,采用轮转的方法,力图使得每一个逻辑文件系统的文件都尽量分散在不同的数据文件系统上。

4.3 NFS 支持

逻辑文件系统的设计目标之一就是为 NFS 服务器提供一个服务器端的高性能文件系统。支持 NFS 的文件系统必须支持一些特殊的接口,包括 export_operations 和 send_file 等。

逻辑文件系统的 export_operations 接口的实现方法比较特殊,因为逻辑文件系统的元数据操作都是使用下层的元数

据文件系统来实现的，所以逻辑文件系统在实现这个接口的时候，采用了 2 阶段操作的方法：第 1 阶段是直接调用元数据文件系统的 export_operations 相应的接口来处理元数据的 dentry 和 file handle 信息；第 2 阶段是通过元数据文件系统进而构造逻辑文件系统的 dentry 和 file handle。

在 Linux2.6 内核中，为了加速 NFS 之类的网络文件系统的读数据性能，引入一个新的接口，叫做 sendfile，它的主要加速作用来自于把文件系统的页缓存直接作为网络发送的数据内存，从而避免一次从页缓冲拷贝到网络缓冲区的内存拷贝。逻辑文件系统为此专门提供了 sendfile 函数。具体的实现方法是，从下层数据文件系统的页缓冲中，直接抓取页面，把这些页面作为 sendfile 的回调参数返回给 NFS，从而实现了在 NFS 下无内存拷贝的高效率读文件操作。

5 逻辑文件系统性能评价

5.1 测试平台配置

测试平台选择为一台配备有多个高性能 IO 通道的服务器，CPU 是一颗双核心的 64 bit AMD Opteron 275，主频 2.2 GHz，内存为 4 GB DDR333。主板集成 3 个 AMD8131 PCI-X 控制器，总共 6 个 PCI-X 通道，共 12 个 PCI-X 插槽。2 块双端口的 320 MB LSI SCSI 卡，每个 SCSI 通道上接 3 块 Seagate 146 GB 的 SCSI 硬盘，总共 12 块硬盘。

操作系统使用 Suse Linux 10.0，内核版本 2.6.13。元数据文件系统选用元数据性能比较好的 reiserfs，数据文件系统选用数据性能比较好的 xfs 文件系统。

5.2 元数据性能

元数据的性能，主要是考察逻辑文件系统对于空文件和目录的创建和删除的操作性能。测试的方法是使用 mdtest[6] 工具，在逻辑文件系统的目录上创建删除大量的目录和文件，然后统计每秒能够完成的操作数，并用这个数据与同一台计算机上直接操作元数据文件系统的性能数据，以及集成了同样数目磁盘的 LVM+xfs 的性能数据进行对比，以此来分析逻辑文件系统的元数据处理性能和处理元数据的开销。逻辑文件系统元数据性能如图 3 所示。

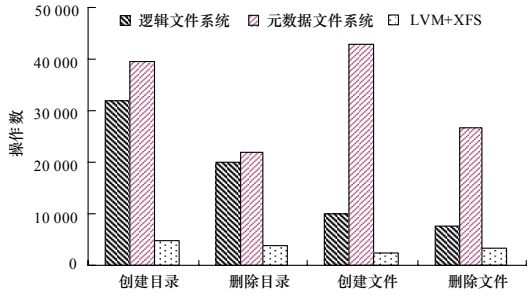

图 3 逻辑文件系统元数据性能

测试参数选择为：创建删除 100 000 个文件和目录，反复执行 5 次，取平均结果。

从测试数据可以看出，逻辑文件系统的元数据性能比在 LVM 上建立的 xfs 文件系统要好很多。在文件的创建和删除操作上，因为逻辑文件系统还要同时创建和删除数据文件系统上的数据文件，所以性能和元数据文件系统有一些差距。

5.3 数据读写性能

数据读写性能主要是通过进行大量的连续的数据读写操作，来测试逻辑文件系统的峰值数据读写带宽。采用的测试方法是用 iozone[7]工具，用多个线程，同时读写大量的连续数据，然后统计所有线程的聚合带宽。作为对比，测试了逻辑文件系统 Single 布局的聚合带宽和 LVM 的聚合带宽。逻辑文件系统数据读写带宽如图 4 所示。

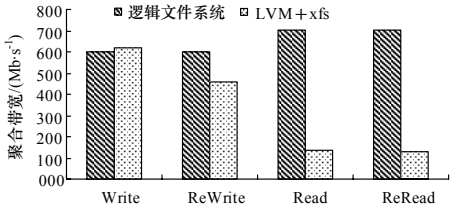

图 4 逻辑文件系统数据读写带宽

测试的参数选择为：12 个线程同时读写，文件大小为 4 GB，读写的块大小为 2 MB。

从测试结果中可以看出，逻辑文件系统在写数据的性能上和在 LVM 上建立的 xfs 文件系统接近，但是在读性能上远远超过了 LVM，证明逻辑文件系统的设计性能优越。

6 结束语

针对目前存储服务器和 Web 服务器等面临的在大量并发文件访问时的 I/O 性能问题，本文提出一种逻辑文件系统技术。它把多个磁盘设备上的物理文件系统虚拟成一个独立的文件系统，集成这些文件系统所在磁盘设备的带宽和容量，为用户提供一个大容量的、高性能的服务器端文件系统。通过集成不同的元数据和数据文件系统，逻辑文件系统综合了各种文件在元数据和数据性能上的优点。经过测试，逻辑文件系统在元数据的操作吞吐率和数据读写的聚合带宽上，都表现出了较好的性能。

参考文献

[1] Patterson D A, Gibson G, Katz R H. A Case for Redundant Arrays of Inexpensive Disks[C]//Proceedings of International Conference on Management of Data. Chicago, Illinois: [s. n.], 1988: 109-116.

[2] Lewis A J. LVM HOWTO[EB/OL]. (2004-10-10). http://www.tldp.org/HOWTO/ LVM-HOWTO/.

[3] Wright C P, Unionfs Z E. Bringing Filesystems Together[J]. Linux Journal, 2004, (128): 24.

[4] Joukov N, Rai A, Zadok E. Increasing Distributed Storage Survivability with a Stackable RAID-like File System[C]// Proceedings of the 2005 IEEE/ACM Workshop on Cluster Security. Stony Brook, NY: [s. n.], 2005: 82- 89.

[5] Sun Microsystems Inc. NFS: Network File System Protocol Specification[S]. RFC 1094, 1989-04.

[6] Loewe W E, Hedges R M, McLarty T T, et al. LLNL's Parallel I/O Testing Tools and Techniques for ASC Parallel File Systems[C]// Proceedings of the 2004 IEEE Cluster Computing Conference. San Diego, CA: [s. n.], 2004-09.

[7] Norcott W D, Capps D. Iozone File system Benchmark[Z]. (2003-10-10). www.iozone.org.