为 EXT3 文件系统增加快照功能

谢全朝[1]，刘日升[1]
[1]大连理工大学计算机系(116023)
E-mail: rcxqc2002@yahoo.com.cn

摘　要：本文分析了 EXT3 文件系统的磁盘存储布局，介绍了快照功能的实现方法。快照功能是完全在文件系统内部实现，没有修改内核接口或者改变文件系统的调用函数，从而对外部的应用程序没有影响。根据数据备份和恢复的需要，介绍了在物理文件系统层增加快照功能的方法。实验结果表明，该快照功能并没有明显降低文件系统的性能。

关键字：EXT3 文件系统　快照　数据备份

1 引言

第三代扩展文件系统（EXT3）[1]是最初为 Linux 操作系统设计的成熟和稳定的文件系统，在大多数 Linux 发布版本中是默认的文件系统，它能够提供给许多用户和工作负载合理的性能和扩展性。EXT3 等常见的文件系统只是提供给用户文件和目录的当前版本，如果用户不小心删除了本来有用的文件，他就会陷入到恢复文件的麻烦中去。如果用户能够保存一个文件和目录的过去几个版本，他就能够容易的恢复数据，也能够看到一个文件的改变历史。

快照技术[2]是实现文件的版本控制技术的一种方式。目前已经有很多的快照文件系统被实现。Wayback[3]是一个用户级的版本控制文件系统。在该文件系统中，用户可以访问一个文件过去的任何版本。版本控制在写操作的时候被自动完成，每个对文件的写操作都创建一个新的版本。缺点是为了实现 Wayback，每个底层的文件系统必须实现版本控制功能，这带来了很多不便。

本文主要根据数据备份的需要，在不需要额外磁盘等硬件的情况下，在 EXT3 文件系统的硬盘存储级别上实现了快照功能。可以在用户删除文件时，不是直接删除文件，而是另外分配数据块，把原来的文件作为快照保存起来，便于恢复数据。测试表明，在修改操作不是太频繁的情况下，保存快照并不需要太大的额外空间。

2 EXT3 文件系统的数据备份原理

2.1 数据备份方法

现在的数据备份方法主要有两种。一种是逻辑备份，备份软件通常既可以进行文件操作又可以对磁盘块进行操作。基于文件的备份系统能够识别文件结构，并拷贝所有的文件和目录到备份资源上。这样的系统跨越了存储在每个 inode 上的指针，顺序的读取每个文件的物理块。然后备份软件连续的将文件写入到备份媒介上。这样的备份使得每个单独文件的恢复变得很快。但是，连续的存储文件会使得备份速度减慢，因为在对非连续存储在磁盘上的文件进行备份时需要额外的查找操作。这些额外的查找操作增加了磁盘的开销，降低了磁盘的吞吐率。基于文件的逻辑备份的另外一个缺点就是对于文件的一个很小的改变也需要将整个文件备份。

与之相比，物理的或“基于设备的备份”系统在拷贝磁盘块到备份媒介上时忽略文件结构。这样会提高备份的性能，因为备份软件在执行过程种，花费在搜索操作上的开销很少。但是，这种方法使得文件的恢复变得复杂而且缓慢。因为文件并不是连续的存储在备份媒介上。为了允许文件恢复，基于设备的备份必须要收集文件和目录是如何在磁盘上组织的信息，才能使得备份媒介上的物理块与特定的文件相关联。因而，基于设备的备份适合于指定一个

特定的文件系统来实现，并且不易移植。

由于本文所需要的恢复操作不多，而且要尽量节省硬盘空间，所以本文采用的是物理备份方式。

2.2 EXT3 文件系统的磁盘布局

现在的操作系统支持多种文件系统，不同的操作系统采用的技术方案各不相同。Linux 内核使用的是虚拟文件系统（VFS）体系结构。虚拟文件系统是一个接口，该接口能够使本地的多种文件系统甚至是远程的文件系统共存于同一台计算机上。虚拟文件系统提供了各种文件系统的公共接口，对于接口的操作对系统和用户都是透明的。当应用程序想某个文件系统请求服务时，文件子系统将通过 VFS 调用相应的函数，该函数首先处理一些与设备无关的操作，然后根据 VFS 结构以及它的 inode 数据结构提供的信息，调用真正文件系统种相对应的函数，用这些函数来处理与特定设备相关的操作。

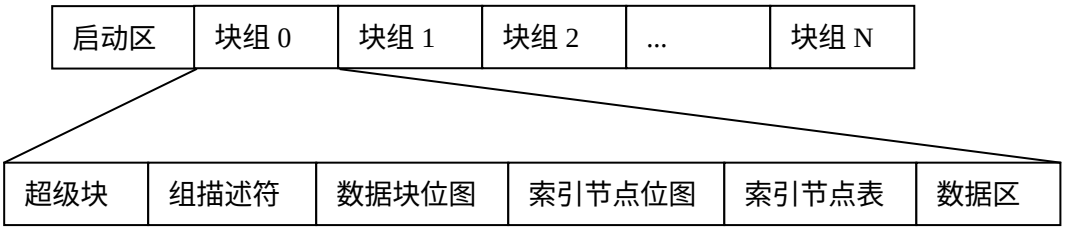

图 1 EXT3 逻辑磁盘的布局

Fig 1 EXT3 logic disk layout

EXT3 文件系统是用来管理和组织保存在磁盘驱动器上的数据的系统软件。文件系统的作用就是在应用概念的文件和存储设备之间提供一个中间层，以使多个文件驻留在一个存储设备上，由文件系统来管理所有文件的存储。除了保存以文件方式存储的数据以外，ext3 文件系统同样存储和管理关于文件和文件系统自身的一些重要信息(例如：日期时间、属主、访问权限、文件大小和存储位置等等)。EXT3 的磁盘布局如图一所示。启动区包含硬盘的硬件信息，不由文件系统管理。其他的部分被分割成多个块组，每个块组有相同的大小并且被顺序存储，因此内核可以仅仅通过它的索引号来得到它在磁盘上的位置。超级块保存文件系统特有的数据，包括文件系统的类型、操作函数集合和主次设备号等。超级块和组描述符在每个块组中都是相同的，只有块组 0 中的超级块和组描述符才被系统使用，其他的备份是为了一致性检查而保留。

2.3 修改 EXT3 的元数据来支持快照功能

我们通过修改 EXT3 的索引节点、目录项和超级块来实现快照功能，没有修改虚拟文件系统的数据结构。这样就不和内核中的页缓冲机制发生联系，从而不改变除了写操作之外的 EXT3 的运行机制。

如图二所示，在 EXT3 文件系统的磁盘索引节点中增加了三个变量来支持快照功能。i_epochnumber(索引节点纪元数)是从纪元（00:00:00 UTC, 1970 年 1 月 1 日）时间以后的秒数，代表快照被建立的时间，这个数字可以通过 gettimeofday 函数来获得。i_cowbitmap 代表快照存在的位图，描述一个文件的磁盘块的信息。i_nextinode 指向同一文件的快照链表中下一个快照的索引节点。

EXT3 文件系统的目录项作为索引节点来实现，其中的数据块包含目录项的内容。目录项的快照和文件的快照按照相同的方式来实现。EXT3 文件系统的超级块增加了一个变量，s_epoch_number，来记录超级块的快照时间。

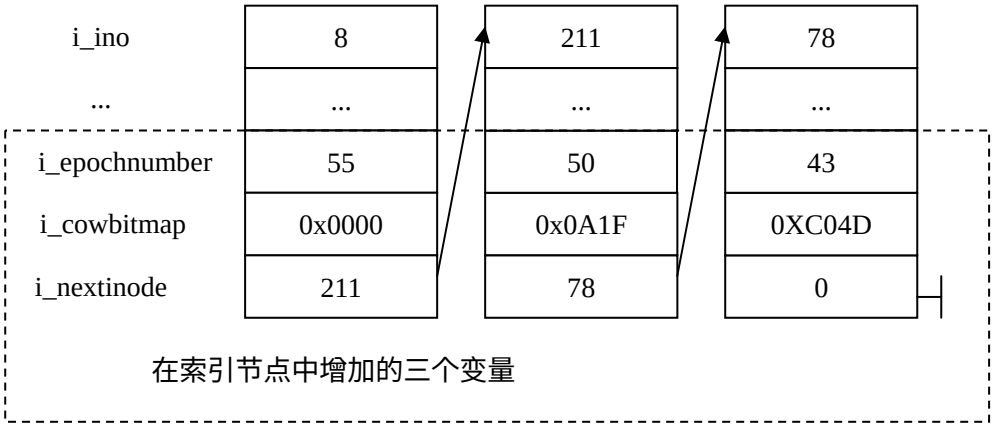

图 2 磁盘和内存索引节点被改进以支持快照和写时复制

Fig 2 disk and memory inodes are revised to support snapshot and copy-on-write

3 快照功能的写时复制实现

本文使用面向磁盘的写时复制机制。数据块的拷贝只在磁盘上而不再内存中存在。这个操作系统中其他的写时复制形式不同，它们创建两个内存拷贝，如创建进程（vfork）和共享页面的虚拟内存管理。我们只在文件的数据变化时需要创建一个新的物理版本。通常，物理版本会和原来的版本共享没有改变的数据。写时复制允许文件的不同版本共享没有改变的数据块（图三）。

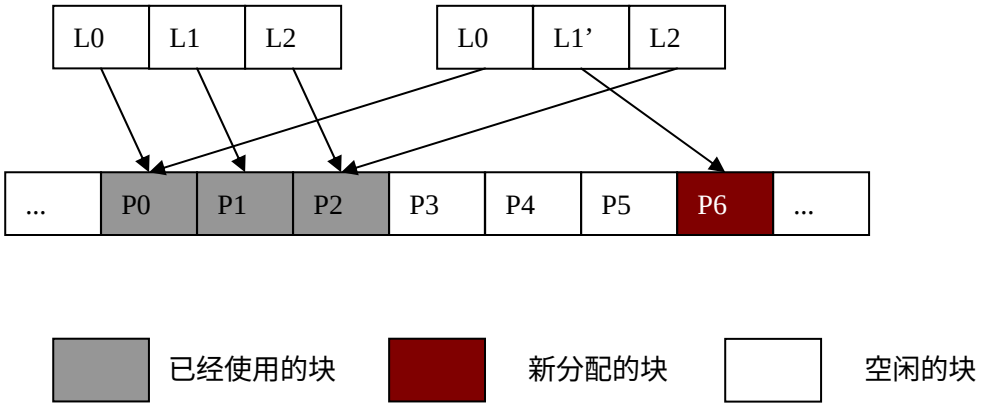

图 3 一个写时复制的例子

Fig 3 an example for copy-on-write

当一个文件的最近的版本先于修改时间时，任何对该文件的改变都将创建一个新的物理版本。第一步是复制索引节点，复制的索引节点共享所以公用的数据块。这包括所有的间接块，也包括二重和三重间接块。在新文件的一个逻辑块被第一次更新时，EXT3 分配一个新的物理磁盘块来存放数据，为旧版本保存旧数据块的一个拷贝。后来的在同一时间点对相同数据的更新被写在合适的地方；写时复制发生在大多数的时间点。对间接块（如，二重和三重间接块）数据的更新，不但改变数据块，也改变间接块。EXT3 在一个间接块被写时复制时分配一个新的磁盘块。

3.1 内存管理

为了实现写时复制，我们隔离那些磁盘文件系统的函数，我们不侵入到内核组件如 VFS 或页缓冲。为了达到这种隔离，我们在内存中平衡 Linux 的多个接口，存有文件数据的内存页被存放在文件系统缓冲中，这些缓冲把内存块映射到磁盘块。在 Linux 中，在更新内存中的一个页之前，VFS 向磁盘文件系统传递一个 write 系统调用。这允许一个文件系统把文件偏移量映射到一个内存地址，并在需要时把数据从缓冲区写到磁盘。我们在此时决定是否分配一个新的磁盘块。我们替换在缓冲区中一个已经存在的磁盘块，并标记缓冲区为脏的，然后 write 调用继续使用相同的内存页。在未来的某个时间点，缓冲管理器把脏的块作为缓冲管理的一部分写到磁盘上。真正的拷贝是在此时被创建。通过重新分配，我们创建文件系统块在磁盘上的拷贝，而不是在内存中拷贝数据。对数据块来说，写时复制不引发另外的 I/O，因为脏的缓冲被 write 调用更新，需要被以某种方式写回磁盘。

3.2 写时复制状态位图

我们把位图嵌入到 EXT3 的索引节点和间接块中，这些索引节点和间接块允许系统记录那些被写时复制执行的块。在索引节点中，我们用一个位代表每个直接块、间接块、二重间接块和三重间接块。值为零的位代表一个新的块在下次写时需要被分配，值为一的位代表在这个时间点上一个新的分配块已经被执行，数据将在合适的地方被更新。EXT3 在复制一个索引节点时将清空整个位图。与之相似，在一个间接块(分别是二重或三重间接块)中，最后的 8 个 32 位包含一个每个位代表被间接块引用的块的位图。当创建这个间接块的写时复制版本时，清空这个位图。因为位图在新的分配发生时被清空，当快照被创建时，什么都不需要做。位图的设计允许位图只有当数据被写时才被更新。

位图的设计允许文件系统当截短一个文件时可以提高性能。截短一个文件是常见的系统操作：当重新写一个文件时，第一步，应用程序经常把一个文件的长度截短为零。截短时，EXT3 重新分配文件所有的块。相反，我们仅仅重新分配在当前时间点被写的块，其他的块仍然保持原样。因此，我们跳过任何相关状态位图等于零的块的分配。对间接块（分别是二重和三重块），我们跳过对应位是零的整个子树的重新分配。通过这种方式，我们减小重新分配块带来的 I/O，在截短时更新自由空间的位图。

4 性能测试

为了确定快照的性价比，我们进行了一系列的实验来比较新的文件系统和原来的 EXT3 文件系统。本次测试采用 LTP（Linux test project）[5]的内核压力测试脚本来衡量操作是否正确和评价性能。基本的测试操作有 9 个部分，每个部分测试一个单独的系统调用。按照顺序，它们是：(1) 创建 5 个层次 62 个目录的 155 个文件，(2) 删除这些文件，(3) 150 个 getcwd 调用，(4) 1000 个 chmods 和 stats，(5) 读写 10 次一个 1048576 字节的文件，(6) 使用 readdir 在一个目录中创建和读 200 个文件，(7) 创建 10 个文件，重命名和对新旧两个检查状态，(8) 创建和读 10 个符号链接，最后，(9) 执行 1500 个 statfs 调用。在 P3 1.8G 内存 256M 的 PC 机上进行。

表 1　每个操作的平均时间

Table 1 average time of various operations

测试操作	Ext3	带有快照功能的 ext3
Create	46.67ms	46.91ms
Remove	6.89ms	3.54ms
Lookup	0.95ms	0.95ms
Attribute	7.24ms	7.38ms

Read	71.28ms	71.91ms
Readdir	15.13ms	15.11ms
Rename	20.28ms	21.11ms
Readlink	5.04ms	6.57ms
Statfs	105.91ms	105.22ms

5 结论

本文在不添加额外硬件的情况下，在 Linux 的 EXT3 文件系统内部实现了一种有效的数据备份的方法。这种方法可以防止由于用户的失误等造成的数据丢失，并能实现数据恢复。未来的工作包括开发一个用户工具可以让用户备份特定的文件，还要继续研究文件的存放策略，提高数据恢复的速度。

参考文献

[1] R. Card, T. Y. Ts'o, and S.Tweedie. Design and implementation of the second extended file system. In Proceedings of the 1991 Amsterdam Linux Conference, 1994.

[2] A. Chervenak, V. Vellanki, and Z. Kurmas. Protecting file systems: A survey of backup techniques. In Proceedings of the Joint NASA and IEEE Mass Storage Conference. March 1998.

[3] B. Cornell, P. Dinda and F. Bustamante. Wayback: A User-level Versioning File System for Linux. The 2004 USENIX Annual Technical Conference, FREENIX track, June, 2004.

[4] LTP Project. http://ltp.sourceforge.net. 2006.

Adding snapshot function to EXT3 file system

Xie Quanchao[1], Liu Risheng[1]

1 Department of computer science Dalian Univ. of Tech. Liaoning Dalian, 116023

Abstract

This paper analyzes EXT3 file system's external disk storage distribution; introduces methods to implement snapshot function. Snapshot function is implemented in the EXT3 file system without changing kernel interface or functions of the file system, so it doesn't infect applications outside. We introduce methods to add snapshot function to physical file system layer based on the need of data backup and restore. The test indicates that adding snapshot function to EXT3 file system does not reduce its performance.

Keywords: EXT3 file system snapshot data backup

作者简介:谢全朝（1981--），研究生；主要研究方向是实时操作系统，嵌入式系统。
刘日升（1944--），教授；主要研究方向是实时操作系统，软件工程等。