

NTFS 文件系统剖析

居锦武, 王兰英

(四川理工学院 计算机科学系, 四川 自贡 643000)

摘要: 针对 Windows 操作系统所使用的 NTFS 与 FAT 文件系统的簇、容量、安全、字符集等, 分析了两种文件系统的区别与优缺点。分析了 NTFS 文件系统使用的数据结构, 特别针对其主控文件表 MFT、MFT 记录、文件结构、目录结构进行了深入的剖析, 给出了 NTFS 文件系统中文件与目录的 MFT 记录结构。

关键词: NTFS 文件系统; 主控文件表; 属性; 记录; 文件; 目录

中图分类号: TP316.86 文献标识码: A 文章编号: 1000-7024(2007)22-5437-03

Analysis of NTFS file system

JU Jin-wu, WANG Lan-ying

(Department of Computer Science, Sichuan Science and Engineering College, Zigong 643000, China)

Abstract: Aimed at the NTFS and FAT's cluster, capacity, security and charset under the Windows, The differences, advantages and disadvantages between the two systems and data structure used in the NTFS are analyzed, especially conducting the detailed research on the MFT, MFT record, file structure and catalogue structure. The file and catalogue's MFT record structure in the NTFS are given.

Key words: NTFS file system; master file table; attribute; record; file; directory

0 引言

Windows NT 最初支持 3 种不同的文件系统 FAT^[1]、HPFS 和 NTFS。NTFS 是 NT 的内定文件系统, 同时 NT 支持 FAT 和 HPFS 以提供对其它操作系统的兼容(虽然 NT4.0 后不再支持 HPFS)。微软公司对 NTFS 的设计目标是要克服另外二个 NT 兼容文件系统的限制, 同时还要能提供高性能, 以满足企业级应用对操作系统的要求^[2]。

本文对 NTFS 与 FAT 文件系统进行比较分析, 详细剖析 NTFS 磁盘数据结构的设计。

1 NTFS 的特性

Windows NT 和 NTFS 是同时开始设计的。在 NTFS 文件系统的基本结构和各部分的功能确定后, 微软公司决定使用 NTFS 作为 NT 的文件系统。NTFS 作为一个全新的文件系统, 它能克服 PC 原有文件系统的限制同时又满足 NT 企业用户的预期需要。

(1) NTFS 的明显特点是它能够很好的适应不断扩大的磁盘容量需求。所有的文件系统都把磁盘分割为逻辑的分配单元, 也就是簇。FAT 文件系统使用 16 位的入口来引用簇, 因此 FAT 最多能寻址 2^{16} 或 65 536 个簇。簇的大小能依据磁盘的大小而改变, 但是大的簇将形成磁盘碎片, 或在簇内部形成

较大的浪费。一个 250 字节的文件仍然要分配一个簇的空间, 如果簇为 16 K 字节, 则 15 K 以上的空间都被浪费了。由于只能寻址 65 536 个簇, 一个簇为 1 KB 的 FAT 磁盘只能寻址 65 MB 空间。要寻址 4 GB 或更大的磁盘则需要 64 KB 的簇, 但它又将产生大量空间浪费。NTFS 采用 64 位地址来引用簇。因此, 如果使用 512 字节的簇, NTFS 能寻址 $2^{64} \times 512$ 字节的磁盘, 在很长的时间内都能满足需要。

(2) NTFS 使用与 NT 一样的安全模型, 而 FAT 和 HPFS 的开发者则忽视了文件系统安全问题。通过使用自由访问控制和系统特权访问控制, 来控制能在一个文件上进行的操作。同时, 文件上的任何操作都将记录在日志中, 日志文件则以 NT 内部格式储存在 NTFS 文件系统内。这种方式使 NTFS 的安全管理与 NT 的安全管理天衣无缝的结合在一起。

(3) 在名字字符集方面。FAT 文件系统使用 8 bit 的 ASCII 码作为文件和目录名字方案的字符集, 这使得 FAT 文件系统的名字局限于英文字符和一些符号。NT 和 NTFS 都使用 16 bit 的 Unicode 码作为名字字符集, NTFS 的这一特性让全球的 NT 使用者都能使用本地语言处理他们的文件。

(4) 在 FAT 文件系统中, 文件是以基于单位的方式储存数据模型。NTFS 则允许在文件内部使用数据流。NTFS 的未命名的数据流等同于传统 FAT 中的文件数据视图, 但 NTFS 的命名文件流则能够代替 FAT 的文件数据单位。

收稿日期: 2006-12-15 E-mail: jjwmail@163.com

基金项目: 四川省教育厅表青年基金项目 (2003A180)。

作者简介: 居锦武 (1976 -), 男, 上海人, 硕士, 讲师, 研究方向为计算机体系结构、嵌入式计算机、设备驱动程序; 王兰英 (1973 -), 女, 四川遂宁人, 硕士, 讲师, 研究方向为计算机体系结构、数据库、Web 应用。

(5) FAT 没有提供对故障-容错的支持。如果在创建或更新文件和目录的时候,系统崩溃了,FAT的磁盘结构可能变得不一致。这种情形可能造成已修改数据的丢失,也可能造成驱动器的损坏和磁盘上数据的丢失。为了解决这一问题,NTFS内建一个事务日志系统,对于任何将要进行的修改操作,NTFS都要在一个特殊的日志文件中做登记^[3]。如果系统崩溃,NTFS检查日志文件并利用它使磁盘恢复到数据损失最小的一致状态^[4-5]。

2 NTFS的磁盘管理

用来创建NTFS磁盘分区和逻辑驱动器的工具能选择簇的大小。程序基于磁盘的大小来决定簇的大小,并且能够在小的簇所带来的管理开销与大的簇所带来的空间浪费之间做出选择。表1显示了缺省的簇的大小。

表1 缺省的簇尺寸

卷尺寸	簇尺寸
512 MB	512 bytes
513 MB~1024 MB	(1 GB)1 KB
1025 MB~2048 MB	(2 GB)2 KB
2 049 MB	4 KB

除了储存用户的文件和目录数据外,NTFS将一些用于磁盘管理的数据也储存在文件中,存储在文件中的与文件系统相关的数据和用户文件与目录中存储的数据都叫做元数据^[6]。当你初始化一个NTFS磁盘时,NTFS将创建11个元数据文件。

表2列出元数据文件和功能描述。浏览NTFS磁盘时,这些文件通常是看不见的,这些文件以\$作为文件名的首字母。在命令行方式下,用命令dir/ah,能看到这些文件的信息。

表2 NTFS中MFT的元文件

文件名	记录号	功能描述
\$MFT	0	主控文件表自身,NTFS的控制中
\$MFTMIRR	1	MFT前16个记录的备份
\$LOGFILE	2	事务日志文件
\$VOLUME	3	卷属性定义表
\$ATTRDEF	4	属性定义表
.	5	卷的根目录
\$BITMAP	6	记录卷中簇的分配情况的位图
\$BOOT	7	引导文件
\$BADCLUS	8	坏簇文件
\$SECURE	9	安全文件
\$SUPCASE	10	MFT自身的扩展文件记录

NTFS除了要将所完成的操作登记在日志文件中,以避免磁盘数据的损失,它还使用信号机制来保护它自己在磁盘上的数据结构。如果读数据时发生错误,NTFS把被读的簇标识为一个坏簇,再将数据映射到磁盘的另外一个位置,然后更新元文件\$BADCLUS,避免以后再次使用该簇。如果存在故障-容错功能的磁盘驱动器,它们将返回信息给NTFS文件系统,利用它们的热修复能力保护其它的数据。

\$BITMAP文件是一个非常大的位数组,数组中的每个位对应着磁盘上的一个簇。如果某位为0,说明对应该位的簇还

未使用,否则簇正在使用中。这个文件由NTFS维护,它跟踪着磁盘上所有的未使用簇,如果要为文件分配新的空间,将从这些未使用的簇中分配。

2.1 主控文件表

NTFS的控制中心是主控文件表MFT^[7]。MFT映射了磁盘上所有文件和目录,包括NTFS的元数据文件,所以MFT类似于FAT文件系统的文件分配表。MFT以记录为单位进行划分。在大部分MFT记录中,储存着描述磁盘上文件或目录的特性(安全属性和其它的属性,如只读或隐藏等)和它在磁盘上的位置信息的元数据。MFT本身也是一个由NTFS用MFT中的记录映射的文件。这样的结构使MFT很容易扩展,使NTFS能很有效率地追踪现存的元数据数量,来决定它们所占用的空间。

NTFS在内部使用MFT中的记录来标识文件和目录,这些记录描述了这些元数据的开始位置信息。例如,表2中所列出的元数据文件都在MFT中有一个预先分配的开始(基本的)记录,在第二列中显示这个值的情况。记录的尺寸通常是1KB,也可能更大些。\$MFTMIRR文件是NTFS预防数据丢失的另一个手段。\$MFTMIRR文件中包含MFT的最初16个记录的一份拷贝,而且NTFS将其储存在磁盘的中央位置,MFT则储存在磁盘的开始位置。如果NTFS在读MFT时发生问题,它就引用它的副本\$MFTMIRR文件。NTFS磁盘的启动记录(磁盘开始的512字节)包含着MFT和\$MFTMIRR文件的位置信息。

MFT的存取效率对一个NTFS磁盘的总体性能具有重要的影响,为此,NTFS要能快速访问MFT。因为MFT也是文件,其大小变化可能使它产生碎片。如果NTFS不能预先连续的分配整个MFT,就会导致碎片的产生。当MFT变大而其它的文件使用的簇正好在它原来的结束处之后,MFT必须在磁盘中找到另一处未使用的空间,这样就导致MFT文件所映射的簇的顺序不连续了。

碎片化的MFT意味着需要多次磁盘操作才能读完一个记录,从而导致较低的性能。为了避免MFT碎裂,NTFS在MFT周围保留一定范围的簇不分配给其它的文件和目录。这样,在MFT需要增长时,能增加找到连续簇的机会,也能尽量少的使MFT靠近其它的数据。当保留区域外的磁盘空间变得不足时,区域中的簇也能分配给其它用途。因此,在磁盘剩余容量较小时,MFT变得破碎的危险将会增加。NTFS不允许磁盘碎片整理工具整理MFT所占用的簇。

2.2 MFT中的记录

MFT中的记录由一个包含关于记录的基本信息的小表头和紧跟着的一些属性组成,这些属性用于描述记录所对应的文件或目录的数据或特性。

图1显示了MFT记录的基本结构。

记录的表头包括NTFS用于完整性检验的序号、指向记录的第1个属性的指针、指向记录的第一个自由字节的指针。如果记录不是文件的第1个记录,还包括文件的基本记录在MFT中的记录号。

NTFS使用属性储存所有的文件和目录信息。表3列出了常用的属性类型。在磁盘上,属性被分为2个逻辑部分:属

MFT 记录			
表头	属性	属性表头	自由空间

图 1 MFT 记录的基本结构

表 3 NTFS 常用的属性

属性类型	属性内容描述
\$VOLUME_VERSION	卷的版本
\$VOLUME_NAME	卷的标识
\$VOLUME_INFORMATION	卷的状态信息等
\$FILE_NAME	存放文件/目录名
\$STANDARD_INFORMATION	文件的时间相关属性
\$SECURITY_DESCRIPTOR	安全描述
\$DATA	文件数据
\$INDEX_ROOT	B+树的根结点
\$INDEX_ALLOCATION	B+树的所有子结点
\$BITMAP	目录内容映像
\$ATTRIBUTE_LIST	描述非常驻属性的表头
\$SYMBOLIC_LINK	未使用
\$EA_INFORMATION	OS/2 兼容的扩展属性
\$EA	OS/2 兼容的扩展属性

性表头和属性数据。表头储存属性的类型、名字和标志,另外还记录着属性数据的位置。NTFS 使用了一个巧妙的优化性能的设计:只要有可能,它将在 MFT 记录里面直接储存属性数据,而不是在磁盘的其它地方分配簇来储存属性数据。存储在 MFT 记录中的属性称为常驻属性,其它的称为非常驻属性。只有当记录表头、属性表头、属性数据都能放入一个记录时,属性才能是一个常驻属性。因此,1 KB(通常的 MFT 记录大小)是 NTFS 磁盘上的属性数据长度的上限。如果属性的数据是常驻的,属性表头将指向在 MFT 记录中的数据的位置。显然,不论属性是常驻还是非常驻的,文件名、标准信息,以及安全属性总是驻留在 MFT 记录中的。如果 NTFS 必须在 MFT 的外部储存属性的数据,则该属性储存在 MFT 记录中的属性表头将包含在磁盘上寻址属性数据的信息。对整个系统中文件和目录的统计表明,小于 1 KB 的文件和目录占相当大一部分,NTFS 的这种设计能够显著提高系统的性能。

为属性数据分配的存储空间称为运行。运行像大多数的 NTFS 数据结构一样,有一个表头用来标识映射到运行中的属性数据的簇。由于属性和大量的数据可能跨越多个 MFT 记录,而运行要覆盖文件的不同部分,采用这种方式就显必不可少。运行由多个运行项组成,一个运行项包含一个虚拟簇号(VCN),它是属性数据内部簇的顺序编号;一个逻辑簇号(LCN),它是数据储存在磁盘上的位置;以及数据在磁盘上所占用空间的连续簇的数量。

如果一个文件有太多属性要填入 MFT 记录中,NTFS 将分配附加的记录并在基本记录中储存一个属性-列表属性。属性列表指向在附加记录中的属性的位置,列表的每一项对应着一个属性。如图 2 所示,Mark.txt 是较大而分成碎片的文件,它需要文件名属性,数据属性,和标准信息属性。它也有一个安全属性,为简单起见忽略了。MFT 将会包含一个或更多的记录,每个记录包含属性的表头,以及其它一些属性数据。名

字和标准信息是常驻的,并在文件的第一个 MFT 记录里面储存。数据属性是非常驻的,被分离为两个属性表头和两个记录中的运行。第 1 个运行是磁盘中簇号 200 开始的 2 个簇,第 2 个运行则是开始于 1033 的 4 个簇。属性列表属性包含一个指向第 2 个 MFT 记录所存储的数据属性的表头。

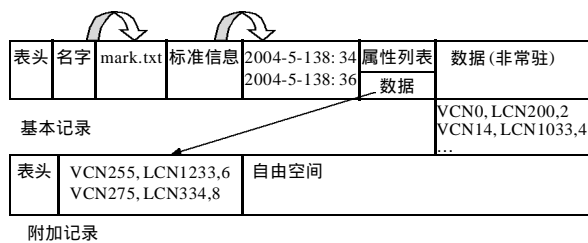


图 2 文件的 MFT 记录结构

2.3 目录结构

NTFS 目录由属性的索引组成,NTFS 使用索引属性对照文件名字,一个目录项中包含着文件的名字和文件的标准信息(时间相关信息)属性的一份拷贝。因此 NTFS 不需要读文件的 MFT 记录就能显示目录信息,这种方式对目录浏览操作提供了较高的性能。

当目录项目数据填满了一个 MFT 记录,一个根索引属性类型用来描述记录中项的位置。如果目录的内容增长,用于描述它的项可能超过文件的 MFT 记录长度。NTFS 分配索引缓冲区去存储附加的项。索引分配属性的表头记录了缓冲区的位置。这些缓冲区通常为 4 KB,包含在其中的目录项是可变的,因为这些目录包含文件的名字,而文件名是可变的。为了尽可能有效率的进行文件的查询,NTFS 在根索引和索引缓冲区内进行预排序,排序使用的数据结构是 B+树。

图 3 显示一个目录的结构。为简单起见,目录中只包含少数几个项,但目录项被分离进入根索引和两个索引缓冲区。实线的箭头显示 NTFS 完成一个目录查询时的扫描方向。虚线的箭头显示索引分配属性中的运行如何引用两个缓冲区。

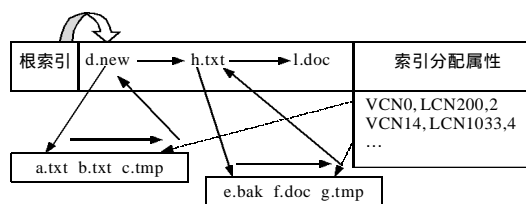


图 3 目录的 MFT 记录结构

3 结束语

通过以上对 NTFS 文件系统的基本结构的分析,可见,NTFS 文件系统的这种组织结构和 FAT/FAT32 文件系统有很大的区别与 Linux 操作系统使用的 EXT2 文件系统^[8]也有很大的区别,这样设计的优点是便于文件系统对数据的定位和处理,便于通过文件的安全描述符对文件进行保护,便于在系统故障时,快速实现对数据的恢复,也很容易实现对卷的扩展。相信随着信息技术的发展和 NT 系列操作系统应用的普及,针对 NTFS 文件系统的开发会有更广泛的应用。

(下转第 5460 页)

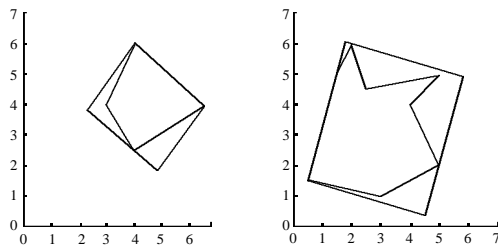


图5 面积最小的矩形包络

图6、图7这两条曲线分别是求解矩形包络时用遗传模拟算法(SA&GA)和GA算法求得的多边形与矩形包络面积比率和迭代次数关系图,曲线表明算法都是收敛的。两张图都表明遗传模拟算法(SA&GA)在收敛速度上要比GA算法的收敛的速度要慢但形成的矩形包络的面积也比GA算法的产生的矩形包络的面积小。图8中的3条曲线是遗传模拟算法(SA&GA)与QPSO算法,GA算法相比,在收敛速度和最后的优化结果上来看,遗传模拟算法(SA&GA)比先进的QPSO算法在性能上还是有所差距的。但不影响其的实用价值。

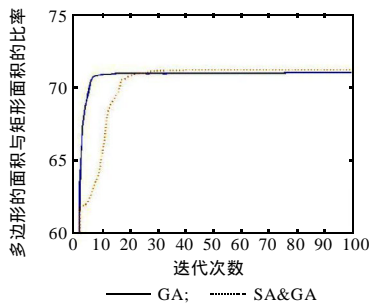


图6 多边形1与矩形面积的比率和迭代次数关系

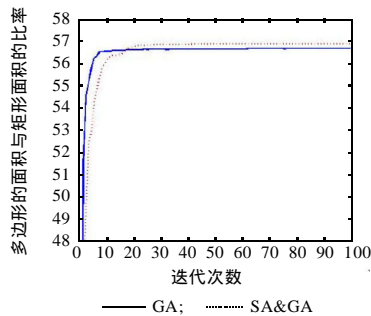


图7 多边形2与矩形面积的比率和迭代次数关系

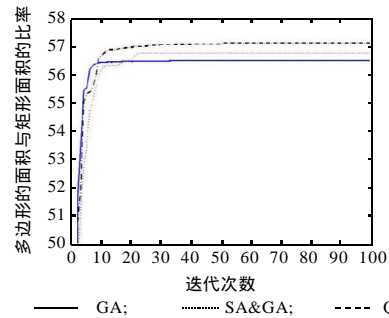


图8 多边形2与矩形面积的比率和迭代次数关系

5 结束语

矩形包络问题无论是在实际工程应用中还是在科学研究中都有十分重要的意义,本文对遗传模拟算法(SA&GA)求解矩形包络问题进行了研究,并比较了基于遗传模拟算法(SA&GA)、QPSO算法和GA算法求解矩形包络问题的结果,比较表明在求解矩形包络问题时遗传模拟算法(SA&GA)的收敛能力优于GA算法,虽然速度略低于GA算法。由此表明遗传模拟算法(SA&GA)在求解矩形包络问题领域具有较高的使用价值。

参考文献:

- [1] Sun J, Xu W B. A global search strategy of quantum-behaved particle swarm optimization[C]. Proceedings of IEEE Conference on Cybernetics and Intelligent Systems, 2004: 111-116.
- [2] Sun J, Feng B, Xu W B. Particle swarm optimization with particles having quantum behavior[C]. USA: Proceedings of Congress on Evolutionary Computation, 2004: 325-331.
- [3] 曾建潮, 介婧, 崔志华. 微粒群算法[M]. 北京: 科学出版社, 2004: 104-105.
- [4] 刘瑞杰, 须文波. 求解矩形件优化排料蚁群算法[J]. 江南大学学报, 2005, 20(10): 23-26.
- [5] 印鉴, 李明. 基于遗传算法的最优布局问题求解[J]. 计算机研究与发展, 2002, 39(10): 1269-1273.
- [6] 张玉萍, 张春丽, 蒋寿伟. 皮料优化排样的有效方法[J]. 软件学报, 2005, 16(2): 316-323.
- [7] 洪灵, 王耕. 一种不规则零件排样的快速解码算法[J]. 计算机辅助设计与图形学学报, 2005, 17(11): 2465-2470.
- [8] 杨彩, 顾海明, 史俊友, 等. 混合遗传算法在矩形件优化排样中的应用[J]. 锻压技术, 2005, 47(3): 6-8.

(上接第5439页)

参考文献:

- [1] 邓波. FAT32 文件系统结构研究[J]. 四川师范大学学报(自然科学版), 2000, 23(6): 665-669.
- [2] David A Solomon. Windows 2000 内部揭秘[M]. 北京: 机械工业出版社, 2001.
- [3] 顾宝根, 顾喜梅. 日志结构的嵌入式文件系统研究[J]. 计算机工程与设计, 2004, 25(6): 915-917.
- [4] 秦航, 徐婕. 一种新的文件系统元数据的检查点容错策略[J]. 计

算机工程与设计, 2004, 25(3): 334-336.

- [5] 吴华, 杨安祺. 分布式文件系统中恢复机制的研究[J]. 微计算机信息, 2006, 22(8): 73-75.
- [6] 梁金千, 张跃. NTFS 文件系统的主要数据结构[J]. 计算机工程与应用, 2003, 39(8): 116-118.
- [7] 王兰英, 居锦武. NTFS 文件系统结构分析[J]. 计算机工程与设计, 2006, 27(3): 418-419.
- [8] 包怀忠. EXT2 文件系统分析[J]. 计算机工程与设计, 2005, 26(4): 1022-1024.