(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 102122284 B
(45)授权公告日 2014.07.02

(21)申请号 201010042692.8
(22)申请日 2010.01.08
(73)专利权人 腾讯科技(深圳)有限公司
地址 518057 广东省深圳市福田区振兴路赛格科技园2栋东403室
(72)发明人 邓立波 陈祎
(74)专利代理机构 广州三环专利代理有限公司 44202
代理人 郝传鑫 潘中毅
(51)Int.Cl.
G06F 17/30(2006.01)

(56)对比文件
CN 101095115 A,2007.12.26,
CN 101414299 A,2009.04.22,
EP 1231549 A2,2002.08.14,

审查员 吴瑶

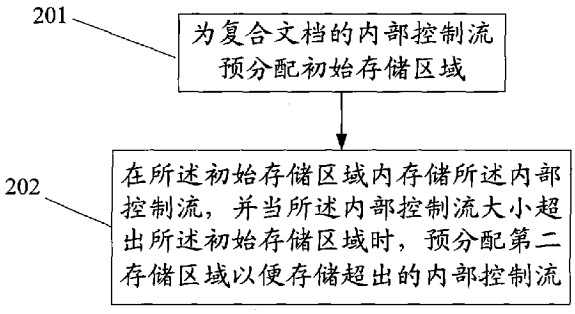
权利要求书3页 说明书10页 附图7页

(54)发明名称

一种复合文档存储、读写方法和装置

(57)摘要

本发明公开了一种复合文档存储、读写方法和装置，该方法包括：为复合文档的内部控制流预分配初始存储区域，所述初始存储区域为连续的扇区或扇区簇；在所述初始存储区域内存储所述内部控制流，并当所述内部控制流大小超出所述初始存储区域时，预分配第二存储区域以便存储超出的内部控制流，所述第二存储区域也为连续的扇区或扇区簇。采用本方法或装置，可减少了复合文档中的用户数据流和内部控制流的碎片。相应的，由于预分配存储空间，使得复合文档中的用户数据流和内部控制流连续存储的概论增加，可引入读缓存和批量写入的策略来优化I/O，提高读写效率。



CN 102122284 B

1. 一种复合文档存储方法，其特征在于，该方法包括：

为复合文档的内部控制流预分配初始存储区域，所述初始存储区域为连续的扇区或扇区簇，所述内部控制流为主扇区分配表、扇区分配表、短流存放流、短扇区分配表及目录流中的一种或多种；

并当所述内部控制流为主扇区分配表、扇区分配表、短流存放流、短扇区分配表及目录流中的多种时，所述为复合文档的内部控制流预分配初始存储区域是指，分别为各种不同的内部控制流预分配不同的初始存储区域；

在所述初始存储区域内存储所述内部控制流，并当所述内部控制流大小超出所述初始存储区域时，预分配第二存储区域以便存储超出的内部控制流，所述第二存储区域也为连续的扇区或扇区簇。

2. 如权利要求 1 所述的方法，其特征在于，所述方法还包括，

当所述超出的内部控制流超过第二存储区域大小时，根据预分配空间策略预分配新的存储区域存储内部控制流；

其中，所述预分配空间策略是指，在存储内部控制流时，当已分配的存储区域不够使用时，总是预分配新的存储区域来存储内部控制流，所有预分配的存储区域都分别为连续的扇区或扇区簇。

3. 如权利要求 2 所述的方法，其特征在于，所述扇区簇是指大小为 8k 字节的连续完整的扇区；所述预分配空间策略中，预分配的初始存储区域大小为 8k 字节，预分配新的存储区域的大小依次为 80k 字节、800k 字节、1M 字节，当前一次预分配的存储区域大小为 1M 字节时，其后预分配的存储区域大小均为 1M 字节。

4. 如权利要求 1 至 3 中任一项所述的方法，其特征在于，所述方法还包括：

为当前正处于打开状态的复合文档，预留数据流存储空间；

在预留的所述数据流存储空间中存储所述复合文档的数据流；

当该复合文档关闭时，释放预留的数据流存储空间中未使用的空间。

5. 如权利要求 4 所述的方法，其特征在于，所述方法还包括：

记录当前处于打开状态的复合文档的预留数据流存储空间的状态；

所述为当前正处于打开状态的复合文档，预留数据流存储空间包括：当需要为复合文档申请预留数据流存储空间时，在扇区分配表中查找没有被使用的扇区或扇区簇，确认该没有被使用的扇区或扇区簇不是已预留的数据流存储空间，将该没有被使用的也不是已预留的数据流存储空间的扇区或扇区簇分配为该复合文档的预留数据流存储空间；

所述当该复合文档关闭时，释放预留的数据流存储空间中未使用的空间还包括：当该复合文档关闭时，删除关于该复合文档的预留数据流存储空间的状态的信息。

6. 一种复合文档读写方法，其特征在于，所述复合文档采用如权利要求 1 所述的方法存储内部控制流，所述复合文档读写方法包括：

在读取内部控制流的扇区或扇区簇时，判断该扇区或扇区簇的相邻的一个或多个扇区或扇区簇中是否也存储了与该内部控制流同类的内部控制流；

若判断结果为是，则将所述扇区或扇区簇以及与其相邻的一个或多个扇区或扇区簇存储的数据一次性读入。

7. 如权利要求 6 所述的方法，其特征在于，所述内部控制流为主扇区分配表、扇区分配

表、短流存放流、短扇区分配表及目录流中的一种。

8. 如权利要求 6 所述的方法，其特征在于，所述复合文档采用预留数据流存储空间存储数据流，所述复合文档读写方法还包括：

在读取数据流时，判断该数据流中是否存在连续的数据块，当判断结果为是时，按最大连续块分批读取该数据流；

在分配和释放扇区链表时，判断相应的内部控制流项是否连续，当判断结果为连续时，对该连续的内部控制流项进行批量操作。

9. 一种复合文档存储装置，其特征在于，该装置包括：

控制流初始预分配模块，用于为复合文档的内部控制流预分配初始存储区域，所述初始存储区域为连续的扇区或扇区簇，所述内部控制流为主扇区分配表、扇区分配表、短流存放流、短扇区分配表及目录流中的一种或多种；

并当所述内部控制流为主扇区分配表、扇区分配表、短流存放流、短扇区分配表及目录流中的多种时，所述控制流初始预分配模块用于分别为各种不同的内部控制流预分配不同的初始存储区域；

控制流第二预分配模块，用于在所述初始存储区域内存储所述内部控制流，并当所述内部控制流大小超出所述初始存储区域时，预分配第二存储区域以便存储超出的内部控制流，所述第二存储区域也为连续的扇区或扇区簇。

10. 如权利要求 9 所述的装置，其特征在于，所述装置还包括，

控制流策略预分配模块，用于当所述超出的内部控制流超过第二存储区域大小时，根据预分配空间策略预分配新的存储区域存储内部控制流；

其中，所述预分配空间策略是指，在存储内部控制流时，当已分配的存储区域不够使用时，总是预分配新的存储区域来存储内部控制流，所有预分配的存储区域都分别为连续的扇区或扇区簇。

11. 如权利要求 10 所述的装置，其特征在于，所述扇区簇是指大小为 8k 字节的连续完整的扇区；所述预分配空间策略中，预分配的初始存储区域大小为 8k 字节，预分配新的存储区域的大小依次为 80k 字节、800k 字节、1M 字节，当前一次预分配的存储区域大小为 1M 字节时，其后预分配的存储区域大小均为 1M 字节。

12. 如权利要求 9 至 11 中任一项所述的装置，其特征在于，所述装置还包括：

数据流预留模块，用于为当前正处于打开状态的复合文档，预留数据流存储空间；

数据流存储模块，用于在预留的所述数据流存储空间中存储所述复合文档的数据流；

数据流预留释放模块，用于当该复合文档关闭时，释放预留的数据流存储空间中未使用的空间。

13. 如权利要求 12 所述的装置，其特征在于，所述装置还包括：

状态记录模块，用于记录当前处于打开状态的复合文档的预留数据流存储空间的状态；

数据流预留模块还用于当需要为复合文档申请预留数据流存储空间时，在扇区分配表中查找没有被使用的扇区或扇区簇，确认该没有被使用的扇区或扇区簇不是已预留的数据流存储空间，将该没有被使用的也不是已预留的数据流存储空间的扇区或扇区簇分配为该复合文档的预留数据流存储空间；

数据流预留释放模块还用于当该复合文档关闭时，删除关于该复合文档的预留数据流存储空间的状态的信息。

14. 一种复合文档读写装置，其特征在于，所述复合文档采用如权利要求 9 所述的装置存储内部控制流，所述复合文档读写装置包括：

扇区判断模块，用于在读取内部控制流的扇区或扇区簇时，判断该扇区或扇区簇的相邻的一个或多个扇区或扇区簇中是否也存储了与该内部控制流同类的内部控制流；

控制流读取模块，用于若判断结果为是，则将所述扇区或扇区簇以及与其相邻的一个或多个扇区或扇区簇存储的数据一次性读入。

15. 如权利要求 14 所述的装置，其特征在于，所述内部控制流为主扇区分配表、扇区分配表、短流存放流、短扇区分配表及目录流中的一种；

所述复合文档采用预留数据流存储空间存储数据流，所述复合文档读写装置还包括：

数据流读取模块，用于在读取数据流时，判断该数据流中是否存在连续的数据块，当判断结果为是时，按最大连续块分批读取该数据流；

控制流项判断模块，用于在分配和释放扇区链表时，判断相应的内部控制流项是否连续；

批量操作模块，用于当判断结果为连续时，对该连续的内部控制流项进 行批量操作。

一种复合文档存储、读写方法和装置

技术领域

[0001] 本发明涉及数据处理领域，尤其涉及一种复合文档存储、读写方法和装置。

背景技术

[0002] 复合文档不仅包含文本而且包括图形、电子表格数据、声音、视频图像以及其它信息。如即时通讯客户端的文件就可以使用复合文档来保存，比如消息记录、表情文件等，随着及时通讯工具使用时间的增长，其相应的复合文档会越来越大。

[0003] 如图 1 所示为一种复合文档的存储 (Storage) 和流 (Stream) 的逻辑结构示意图。复合文档的逻辑结构与文件系统的非常相似，每个文档有一个根存储 (Root Storage)，每个存储下面可以有 0 到多个存储或流。每个存储和流都有一个名字，该名字通常由 16 位的 Unicode 字符构成，最大名字长度为 31 个字符。同一个存储下的存储或流的名字不能相同，不同存储下的存储或流的名字可以相同。

[0004] 复合文档除了头结构以外，所有的数据都以流的形式组织。复合文档的所有流都被分成更小的数据块，叫做扇区。扇区可能包含控制数据或用户数据。整个复合文档包含一个头结构，跟在头结构后面的是一系列的扇区。所有扇区的大小都相同，这个大小值在头结构中设定。

[0005] 扇区以其在文件中的顺序列举，扇区的索引 (从 0 开始) 叫做扇区标识 (SID)，它是一个有符号的 32 位整型值。如果一个 SID 不小于 0，那么它一定指向一个存在的扇区 ；如果 SID 值为负，那么它可能有特殊的含义。

[0006] 构成一个流的所有扇区所形成的链表叫做扇区链，扇区链中相邻的扇区在物理上不一定相邻，为了方便地表示扇区链中各个扇区的相对位置关系，引入了扇区标识链的概念。扇区标识链是一个扇区标识的数组。扇区标识链以流中第一个扇区的扇区标识开始，顺次记录流中扇区的扇区标识，以链表结束 (-2) 结尾。

[0007] 而复合文档中的流按用途分，可以分为内部控制流和用户数据流。内部控制流有目录流，主扇区分配表 (MSAT)，扇区分配表 (SAT)，短扇区分配表 (SSAT)，短流存放流。

[0008] 其中，主扇区配置表 (MSAT，Master Sector Allocation Table) 是一个 SID 数组，顺序指明了用于存放扇区配置表的扇区的 SID。MSAT 的大小等于用于存放 SAT 的扇区的个数，这个值存放在头结构中。

[0009] 扇区配置表 (SAT，Sector Allocation Table) 是一个扇区标识的数组，它包含了所有的用户数据流和内部控制流，SAT 的大小等于整个复合文档中存在的扇区个数。SAT 数组元素的索引就是该元素代表的扇区标识，而元素的值则为该元素代表的扇区在扇区链中的下一个节点。SAT 可能在任意位置包含 Free SID(-1)，这些扇区将不被任何流使用 ；如果该位置包含 End of SIDChain(-2)，表示一个流的结束 ；如果该位置包含 SAT SID(-3)，表示所代表的扇区用于存放 SAT ；如果该位置包含 MSAT SID(-4)，表示所代表的扇区用于存放 MSAT。

[0010] 短流存放流与其他的长度不小于标准流长度的普通用户流一样。它的扇区链中第

一个扇区的 SID 存放在根存储 (Root Storage) 的目录项里。短流存放流的扇区标识链可以从 SAT 中获得。

[0011] 短扇区分配表 (SSAT, Short Sector Allocation Table) 就是一个扇区标识的数组,包含了所有短流的扇区标识链。SSAT 作为一个内部控制流,它的创建过程与普通的流的创建过程一致。SSAT 的第一个扇区标识存放在头结构中。SSAT 作为一个扇区分配表,他的作用与 SAT 极为相似,唯一的区别就是 SSAT 中的扇区标识指向的是短扇区而不是普通扇区。

[0012] 目录流是一个内部控制流,它包含了一个目录项数组,每一个目录项指向复合文档中的一个存储或流,目录流中以 0 为开始的目录项索引称为目录项标识 (directory entry identifier, DID)。

[0013] 在上述的复合文档中没有对扇区分配进行控制,导致大量碎片,I/O 不停的在整个复合文档中跳转,严重影响性能。其中主扇区分配表 (MSAT)、扇区分配表 (SAT)、短扇区分配表 (SSAT)、目录项遍布于整个复合文档中,严重影响复合文档的打开、遍历、读写等性能,而对于流和短流,过小的分配单元和完全没有控制的扇区分配,也造成了大量碎片。

发明内容

[0014] 本发明实施例所要解决的技术问题在于,提供一种复合文档存储、读写方法和装置,以减少复合文档中的碎片。

[0015] 为此,本发明实施例提供了一种复合文档存储方法,包括 :为复合文档的内部控制流预分配初始存储区域,所述初始存储区域为连续的扇区或扇区簇 ;

[0016] 在所述初始存储区域内存储所述内部控制流,并当所述内部控制流大小超出所述初始存储区域时,预分配第二存储区域以便存储超出的内部控制流,所述第二存储区域也为连续的扇区或扇区簇。

[0017] 相应地,本发明实施例还提供了一种复合文档读写方法,所述复合文档采用如前所述的方法存储内部控制流,所述复合文档读写方法包括 :

[0018] 在读取内部控制流的扇区时,判断该扇区的相邻的一个或多个扇区中是否也存储了与该内部控制流同类的内部控制流 ;

[0019] 若判断结果为是,则将所述扇区以及与其相邻的一个或多个扇区存储的数据一次性读入。

[0020] 同时,本发明实施例还提供了一种复合文档存储装置,该装置包括 :

[0021] 控制流初始预分配模块,用于为复合文档的内部控制流预分配初始存储区域,所述初始存储区域为连续的扇区或扇区簇 ;

[0022] 控制流第二预分配模块,用于在所述初始存储区域内存储所述内部控制流,并当所述内部控制流大小超出所述初始存储区域时,预分配第二存储区域以便存储超出的内部控制流,所述第二存储区域也为连续的扇区或扇区簇。

[0023] 本发明实施例还提供了一种复合文档读写装置,所述复合文档采用如前所述的装置存储内部控制流,所述复合文档读写装置包括 :

[0024] 扇区判断模块,用于在读取内部控制流的扇区时,判断该扇区的相邻的一个或多个扇区中是否也存储了与该内部控制流同类的内部控制流 ;

[0025] 控制流读取模块，用于若判断结果为是，则将所述扇区以及与其相邻的一个或多个扇区存储的数据一次性读入。

[0026] 在本发明实施例所提供的方案中，对内部控制流采用预分配存储空间的策略进行存储，减少了复合文档中的内部控制流的碎片。相应的，由于预分配存储空间，使得复合文档中的内部控制流连续存储的概论增加，可引入读缓存和批量写入的策略来优化 I/O，提高读写效率。

附图说明

[0027] 图 1 是现有技术中的一种复合文档的存储和流的逻辑结构示意图；

[0028] 图 2 是本发明实施例中的关于内部控制流的预分配策略下的存储方法的一个流程示意图；

[0029] 图 3 是本发明实施例中的用户数据流的预分配存储空间的存储方法一个流程示意图；

[0030] 图 4 是在图 3 中的存储方法中记录为各用户数据流预留的数据存储空间的状态的一个流程示意图；

[0031] 图 5 是本发明实施例中的复合文档存储装置的一个流程示意图；

[0032] 图 6 是本发明实施例中的复合文档存储装置的另一个流程示意图；

[0033] 图 7 是本发明实施例中的复合文档存储装置的另一个流程示意图；

[0034] 图 8 是本发明实施例中的复合文档存储装置的另一个流程示意图；

[0035] 图 9 是本发明实施例中的复合文档读写装置的一个流程示意图；

[0036] 图 10 是本发明实施例中的复合文档读写装置的另一个流程示意图；

[0037] 图 11 是本发明实施例中的复合文档读写装置的另一个流程示意图；

[0038] 图 12 是用微软复合文档接口打开 db1 的耗时和本发明实施例中的复合文档接口打开 db2 耗时的结果对比示意图；

[0039] 图 13 是用微软复合文档接口模拟读取 db1 中 7 个好友的消息记录和本发明实施例中的复合文档接口模拟读取 db2 中 7 个好友消息记录的耗时对比示意图；

[0040] 图 14 是用微软复合文档接口模拟写入 7 个好友的消息记录和用本发明实施例中的复合文档接口模拟写入 7 个好友消息记录的耗时对比示意图。

具体实施方式

[0041] 下面将结合本发明实施例中的附图，对本发明实施例中的技术方案进行清楚、完整地描述，显然，所描述的实施例仅仅是本发明一部分实施例，而不是全部的实施例。基于本发明中的实施例，本领域普通技术人员在没有作出创造性劳动前提下所获得的所有其他实施例，都属于本发明保护的范围。

[0042] 为了解决现有的复合文档的碎片多导致的文件打开、读写性能差问题，本发明实施例提出了一种完全兼容现有复合文档格式的新复合文档。在本发明实施例中的复合文档中，采用预分配策略（即对内部控制流或 / 和用户数据流预分配一定大小的存储空间，以进行相应的存储）进行复合文档的存储，以保证数据块的连续性，减少碎片的产生；另外，相应于本发明实施例中的复合文档的存储方式，本发明实施例还提出了一种采取缓存读取和

批量写入策略来优化输入 / 输出 (I/O)，从而提升复合文档的打开、读写性能。

[0043] 同时，在本发明实施例中的复合文档可以以扇区为单位进行存储，也可以以扇区簇为单位进行存储。在前期的研究发现，现有的复合文档（如微软复合文档）的 MSAT，目录项，SSAT 每个扇区都不连续。其分配粒度是 512 字节的扇区，分配粒度过小，增加了碎片形成的几率，为从根源上减少碎片，本发明实施例中重新定义一个比较大一点的分配粒度——扇区簇（类似磁盘管理中的簇）。按复合文档的规范，小于 4K 字节 (4×1024 字节，或表示为 4KB) 的即为短流。因此可以选择大于 4K 字节的粒度值，如 8K 字节做为簇的大小。为实现簇分配单元模式，定义了一个非常简单的原则，即每次分配都按照 8K 字节边界对齐分配扇区，这样保证分配粒度始终不小于一个扇区簇。因为簇大小 8K 字节正好是 512 字节的整数倍，所以能完全兼容现有复合文档格式。当然，根据具体的情况扇区簇的大小也可以是其他数值，如为扇区大小的整数倍数值等。

[0044] 如前所述，在本发明实施例中的复合文档中，采用预分配策略来进行复合文档的存储。由于内部控制流和用户数据流的性质的不同，其具体的预分配策略是不同的。如图 2 所示，为本发明实施例中的关于内部控制流的预分配策略下的存储方法，包括：

[0045] 201、为复合文档的内部控制流预分配初始存储区域，所述初始存储区域为连续的扇区或扇区簇。在本实施例，以及本发明的其他实施例中所涉及的存储区域的单位，既可以是扇区，也可以是前述的扇区簇；但，不论是扇区还是扇区簇，在某一具体实施例中，并不会存在采用扇区和扇区簇混合作为存储单位的情况，即在某一复合文档的实现中，要么都是采用扇区作为基本存储单位，要么都是采用扇区簇作为基本存储单位（在同一实施例中，包括内部控制流和后述的用户数据流都采用相同的基本存储单位进行存储）。

[0046] 其中，本实施例中的内部控制流可以是主扇区分配表、扇区分配表、短流存放流、短扇区分配表及目录流中的一种或多种。即可以是对复合文档中的上述的某一种具体的内部控制流（如 MSAT）采用本实施例中的预分配存储区域的方案进行存储，也可以是对复合文档中的多种或全部的内部控制流都采用本实施例中的预分配存储区域的方案进行存储。

[0047] 需要说明的是，若是对复合文档中的多种或全部的内部控制流，所述为复合文档的内部控制流预分配初始存储区域是指，分别为各种不同的内部控制流预分配不同的初始存储区域。

[0048] 202、在所述初始存储区域内存储所述内部控制流，并当所述内部控制流大小超出所述初始存储区域时，预分配第二存储区域以便存储超出的内部控制流，所述第二存储区域也为连续的扇区或扇区簇。

[0049] 在本发明实施例中，由于内部控制流（如 SAT）较大，或是由于复合文件的增长导致 SAT 的增长，可能原来已分配的初始存储区域以及第二存储区域都不够用，则还可以根据预分配空间策略预分配新的存储区域存储内部控制流。其中，所述预分配空间策略是指，在存储内部控制流时，当已分配的存储区域不够使用时，总是预分配新的存储区域来存储内部控制流，所有预分配的存储区域都分别为连续的扇区或扇区簇。

[0050] 并且，在上述实施例中，扇区簇及预分配的存储空间的大小具体可以是，扇区簇是指大小为 8k 字节的连续完整的扇区；预分配的初始存储区域大小为 8k 字节，预分配新的存储区域的大小依次为 80k 字节（即第二存储区域）、800k 字节、1M（兆）字节，当前一次预分配的存储区域大小为 1M 字节时，其后预分配的存储区域大小均为 1M 字节。当然，上述的扇

区簇也可以是 4k 字节的整数倍大小的连续完整的扇区。

[0051] 以下根据内部控制流的具体类型，给出一些进行预分配存储的实施例：

[0052] 1) 主扇区分配表的预分配方法

[0053] 若每个 MSAT 扇区可以容纳 128 个 SAT 扇区的 SID，则每个 SAT 扇区容纳 128 个扇区 SID，这样 1G 字节的文件共需要 64K 字节大小的主扇区分配表。在复合文档的文件头，包含 109 个 MSAT SID。这样即使不使用附加的 MSAT，支持最大 6.8M 字节的复合文档。其中，1G ＝ 1024M ＝ 1024×1024K。

[0054] 主扇区分配表的预分配方法即为，根据 MSAT 的大小及增长情况，不断分配新的存储区域进行存储，如设定初始存储区域大小为 8K，第二存储区域为 80K，其他不断预分配的存储区域大小依次为 80K、800K、1M、1M、1M...。即，对于预分配的存储区域的大小的设定策略可为，对于小于 1M 的按 10 的倍数增长，大于 1M 的则固定为 1M。对于已经使用的预分配的存储区域，把已使用的全部扇区串成 MSAT 扇区链，在 MSAT 扇区中对不存在的 SAT 的 SID 填 -1。对小于 1.28G 的文件，除了文件头外，只有一处 MSAT 碎片，所以可以不考虑合并 MSAT 的问题。

[0055] 2) 扇区分配表的预分配方法

[0056] 按复合文档的格式定义，SAT 的大小与文件大小是直接对应的关系，理论上当 SAT 表增加时，复合文档大小也要增加，但是复合文档中存放的数据流大小可以不变。因为，现有技术中的复合文档（如，微软的复合文档），是没有预分配的概念，它的 SAT 表是随着复合文档大小增加而同时增加了；但是本发明实施例中的复合文档与现有技术中的复合文档不同的是，本发明实施例中的复合文档中的 SAT 表可以预先分配，复合文档大小增加的同时，复合文档中存放的数据大小是可以不增加的。

[0057] 每个 SAT 扇区容纳 128 个扇区 SID，这样 1G 的文件共需要 8M 大小的主扇区分配表。

[0058] 类似 MSAT 的预分配情况，SAT 的存储区域的预分配初始值也可为 8k，以后分配的依次为 80K、800K、1M、1M、1M...，即小于 1M 的按 10 的倍数增长，大于 1M 的固定为 1M。对于小于 800M 的文件，共有 8 个 SAT 碎片。这种模式 SAT 碎片比较少，可以不用实现 SAT 合并。

[0059] 由于 SAT 表中 4 个字节可以指向 1 个数据流的扇区 (512 字节)，如果没有预分配，SAT 占用空间是复合文档的 4/512 即 1/128；本发明实施例中实现的预分配 SAT 的空间最大是当前已经使用的 SAT 空间的 10 倍。因此有，SAT 预分配最大消耗 10/128 ＝ 8％的复合文档空间。

[0060] 3) 短流存放流的预分配方法

[0061] 短流存放流是一个流的容器，用于存放用户所有小于 4K 的流。一般在即时通讯工具中大多数流都属于短流，例如 gif 或 bmp 资源类文件等等，以及小的配置类文件。因此短流存放流的碎片将极大影响复合文档的性能。

[0062] 类似 MSAT 的预分配情况，短流存放流的存储区域的预分配初始值也可为 8k，其余依次为 80K、800K、1M、1M、1M...，即小于 1M 的按 10 的倍数增长，大于 1M 的固定为 1M。若，预分配存储区域按 1M 固定预分配，则每次可以增加 256 个短流，按实际使用经验这个速度足以满足要求。

[0063] 4) 短扇区分配表的预分配方法

[0064] 虽然看上去SSAT与短流存放流大小必须成正比，实际上Windows生成的复合文档中的SSAT可以超出这个数值。这样SSAT不必从512字节起始大小开始增长，其预分配方法和短流存放流的预分配方法一样，也使用模型：按8K、80K、800K、1M、1M、1M... 增长。

[0065] 因为短流存放流同SSAT不需要完全匹配，所以短流存放流扩容的触发点是在每次写入，当容量不够时再扩容（注意扩容后的最大值不能超过与SSAT匹配的尺寸）。

[0066] 5）目录流的预分配方法

[0067] 以微软复合文档为例，其格式定义中每个目录项固定128个字节。

[0068] 目录流的预分配方法使用预分配模型为：初始为8K（可以容纳64个目录项），按8K、80K、800K、1M、1M、1M... 增长，其中对于未用的目录项置为空目录项。这样若少于6400个目录项，只有80K、800K两处碎片，可以不实现合并目录流。

[0069] 在上述各内部控制流的预分配存储方法中，可根据上述的预分配模型在当前预分配容量不够时，进行扩容，如，SAT的扩容触发点是写入数据流容量不够时；MSAT的扩容触发点是随着SAT表增加导致存放MSAT表空间不够时。

[0070] 以上描述了内部控制流的预分配的存储方法，如图3所示，则为用户数据流（或称数据流）的预分配存储空间的存储方法，包括：

[0071] 301、为当前正处于打开状态的复合文档，预留数据流存储空间。即，用户数据流的预分配都采用了预留空间模式，每次分配新的扇区簇或扇区，便预留指定大小的空间，随后的其它对象都不得使用该空间。如，写入一个数据流时，分配一块大小比当前写入的数据大小更大的连续的扇区簇供该流写入，因为当写入该流时，很有可能会稍后再次写入该流；为了2次写入的数据连续，可在写入的时候预留连续的一块空间供下次写入使用。

[0072] 其中，数据流存储空间可以包括普通流存储空间或/和短流存储空间。则部分实施例中的普通流存储空间大小可为相应的当前流长度的50％或以上，所述短流存储空间大小为4k字节。

[0073] 302、在预留的所述数据流存储空间中存储所述复合文档的数据流。

[0074] 303、当该复合文档关闭时，释放预留的数据流存储空间中未使用的空间。

[0075] 由于当前，可能存在多个用户数据流需要进行处理，则需要记录为各用户数据流预留的数据流存储空间的状态，以便当需要预留新的数据流存储空间时参考已预留的数据流存储空间的状态，则上述实施例中还包括：

[0076] 401、记录当前处于打开状态的复合文档的预留数据流存储空间的状态。如，可在打开到关闭复合文档的生命周期中，在内存中建立一张预留空间状态表，保持对该复合文档预留空间状态的跟踪，用于请求空闲扇区簇时的参考。由于可能同时打开多个复合文档，因此在实现上述预留空间状态表时，如，采用代码实现上这个表时，该表可以是一个全局的地图（map），因此，各个复合文档的数据流预留的空间情况各个复合文档是相互知道的。

[0077] 402、当需要为复合文档申请预留数据流存储空间时，在扇区分配表中查找没有被使用的扇区或扇区簇，确认该没有被使用的扇区或扇区簇不是已预留的数据流存储空间，将该没有被使用的也不是已预留的数据流存储空间的扇区或扇区簇分配为该复合文档的预留数据流存储空间。如，当申请新的扇区簇时，首先从SAT中找到没有被使用的扇区簇，同时确认该扇区簇不属于其它对象的预留空间才分配。预留的空间在SAT中不记录扇区链信息。

[0078] 403、当该复合文档关闭时，删除关于该复合文档的预留数据流存储空间的状态的信息。如，当复合文档关闭时，预留空间状态表也同时销毁，不写入任何信息到文件。

[0079] 根据上述实施例中描述的存储方法以下，根据用户数据流的类型描述不同数据流的实现预分配方法的一些实施例：

[0080] 1) 普通流的预分配方法

[0081] 普通流的预分配使用预留空间模式：由于在大多数时候，当前打开并有过写入操作的流只占少部分，这样可以为这部分流预留比较大的空间，每次预留大小定义为当前流长度的50％，从而保持更小的碎片。如果这个预留空间在当前打开期间不能被利用到，下次打开复合文档会被别的流使用。

[0082] 2) 短流的预分配方法

[0083] 按照微软复合文档格式的规定，小于4K的流即为短流。短流的预分配使用预留空间模式：为避免单个短流出现碎片，每个短流不论大小，统一为其预留4K空间。在实现时，按起始地址4K对齐的原则分配扇区，保证每个短流都有4K空间。

[0084] 通过上述描述可知，采用预分配策略对复合文件中的内部控制流和用户数据流进行存储，可以减少复合文档中碎片的产生，并保证数据的连续性。

[0085] 使用本发明实施例中的预分配策略后，复合文件系统里面的数据理论上已经是一大块一大块的，这其中的一些内部控制流，例如SSAT/SAT需要经常访问，反复访问频率会比较高。

[0086] Windows虽然有缓冲预测算法，反复访问的内容会优先加入到缓存，但是由于它不理解文件中数据的具体应用，不知道数据的缓存优先级别，预测算法做不到100％可靠。因此，对于采用前述的实施例中描述的方法存储的复合文档，其读写可以相应的采用下述的方法进行，以进一步提高复合文档的读写效率。本发明实施例中的一种复合文档读写方法中，在读取内部控制流时，包括：

[0087] A1、在读取内部控制流的扇区或扇区簇时，判断该扇区或扇区簇的相邻的一个或多个扇区或扇区簇中是否也存储了与该内部控制流同类的内部控制流。其中，内部控制流为主扇区分配表、扇区分配表、短流存放流、短扇区分配表及目录流中的一种

[0088] A2、若判断结果为是，则将所述扇区或扇区簇以及与其相邻的一个或多个扇区或扇区簇存储的数据一次性读入。

[0089] 如，在本发明实施例中当要读取SAT某个扇区（或扇区簇）时，会先判断一下与其左右相邻的数个扇区（或扇区簇）是否也是存放的SAT(SAT数组中存放的扇区标识有一些特殊标识，比如-3代表该扇区存放的是SAT)，如果是，那就把这一块数据一次性读入。由于I/O消耗主要在磁头跳转，既然花了大量的时间跳到了目的地，就应该多读一点数据。因此，采用上述方案，不替代Windows缓存机制，只是简单的对需要经常访问的数据主动预读取一个大块，而不是按需要每次都读固定的块数，可以极大的提高I/O的效率。

[0090] 另一方面在读取用户数据流时，现有技术中的复合文档，如微软实现的复合文档，按Sector为单位读写，如果读取一大块数据，需要一个一个依次调用SetFilePointerEx和ReadFile，这种方式有几个负面影响：(1) 当读取一个扇区时，系统会预读取后面一大片数据并缓存起来，如果请求的数据块大于系统预计缓存的大小，会发生磁盘寻道（一般硬盘平均寻道时间在10ms以上）。(2)SetFilePointerEx和ReadFile属于内核调用，调用本身

会有消耗，仅在在 Hummer 登录过程中，ReadFile 和 SetFilePointerEx 调用总和超过 4 万次。(3) 频繁 I/O 也会影响到即将开发的事务性文件系统性能。

[0091] 因此，本发明实施例中，在读取数据流时，则采用以下方法：

[0092] B1、在读取数据流时，判断该数据流中是否存在连续的数据块；

[0093] B2、当判断结果为是时，按最大连续快分批读取该数据流。

[0094] 因为数据流预留空间策略保证了流数据的连续性，因此本发明实施例中的复合文档的读取缓存策略是尽量缓存当前读取流的数据；而现有技术中的复合文档的读缓存只是缓存了当前扇区的后面一部分扇区的数据，同时，由于有技术中的复合文档碎片多，这部分数据极有可能并不是当前流的数据。因此，采用上述本发明实施例中的读取数据的方法，可以解决上述现有技术中的问题。

[0095] 另一方面，由于分配和释放扇区链表时需要频繁写入 / 擦除 SAT/SSAT 项，在本发明实施例中的流的数据块是成块连续的，相应的 SAT 表项也是连续的，因此在操作 SAT 表项时可采用如下步骤：

[0096] C1、在分配和释放扇区链表时，判断相应的内部控制流项是否连续；

[0097] C2、当判断结果为连续时，对该连续的内部控制流项进行批量操作。

[0098] 相应于上述各方法的实施例，本发明实施例也提供了相应的实现装置，如图 5 所示，为相应的复合文档存储装置 1，该装置包括：

[0099] 控制流初始预分配模块 10，用于为复合文档的内部控制流预分配初始存储区域，所述初始存储区域为连续的扇区或扇区簇。

[0100] 控制流第二预分配模块 12，用于在所述初始存储区域内存储所述内部控制流，并当所述内部控制流大小超出所述初始存储区域时，预分配第二存储区域以便存储超出的内部控制流，所述第二存储区域也为连续的扇区或扇区簇。

[0101] 其中，内部控制流为主扇区分配表、扇区分配表、短流存放流、短扇区分配表及目录流中的一种或多种；并当所述内部控制流为主扇区分配表、扇区分配表、短流存放流、短扇区分配表及目录流中的多种时，所述控制流初始预分配模块 10 用于分别为各种不同的内部控制流预分配不同的初始存储区域。

[0102] 如图 6 所示，该装置 1 还可包括：

[0103] 控制流策略预分配模块 14，用于当所述超出的内部控制流超过第二存储区域大小时，根据预分配空间策略预分配新的存储区域存储内部控制流；其中，所述预分配空间策略是指，在存储内部控制流时，当已分配的存储区域不够使用时，总是预分配新的存储区域来存储内部控制流，所有预分配的存储区域都分别为连续的扇区或扇区簇。

[0104] 其中，在部分实施例中，扇区簇是指大小为 8k 字节的连续完整的扇区；预分配的初始存储区域大小为 8k 字节，预分配新的存储区域的大小依次为 80k 字节、800k 字节、1M 字节，当前一次预分配的存储区域大小为 1M 字节时，其后预分配的存储区域大小均为 1M 字节。

[0105] 如图 7 和 8 所示，该装置 1 还可包括：

[0106] 数据流预留模块 11，用于为当前正处于打开状态的复合文档，预留数据流存储空间；

[0107] 数据流存储模块 13，用于在预留的所述数据流存储空间中存储所述复合文档的数

据流；

[0108] 数据流预留释放模块 15，用于当该复合文档关闭时，释放预留的数据流存储空间中未使用的空间。

[0109] 状态记录模块 17，用于记录当前处于打开状态的复合文档的预留数据流存储空间的状态。若包括状态记录模块 17 则数据流预留模块 15 还用于当需要为复合文档申请预留数据流存储空间时，在扇区分配表中查找没有被使用的扇区或扇区簇，确认该没有被使用的扇区或扇区簇不是已预留的数据流存储空间，将该没有被使用的也不是已预留的数据流存储空间的扇区或扇区簇分配为该复合文档的预留数据流存储空间；数据流预留释放模块 15 还用于当该复合文档关闭时，删除关于该复合文档的预留数据流存储空间的状态的信息。

[0110] 在图 7 和图 8 中虚线表示的模块和连线，表示该复合文档存储装置可以包括该模块也可以不包括。

[0111] 其中，在部分实施例中，数据流存储空间包括普通流存储空间或 / 和短流存储空间，所述普通流存储空间大小为相应的当前流长度的 50％或以上，所述短流存储空间大小为 4k 字节。

[0112] 相应的如图 9 ～ 11 所示，为本发明实施例中的复合文档读写装置 2，其包括：

[0113] 扇区判断模块 20，用于在读取内部控制流的扇区或扇区簇时，判断该扇区或扇区簇的相邻的一个或多个扇区或扇区簇中是否也存储了与该内部控制流同类的内部控制流；

[0114] 控制流读取模块 22，用于若判断结果为是，则将所述扇区或扇区簇以及与其相邻的一个或多个扇区或扇区簇存储的数据一次性读入。

[0115] 数据流读取模块 24，用于在读取数据流时，判断该数据流中是否存在连续的数据块，当判断结果为是时，按最大连续快分批读取该数据流。

[0116] 控制流项判断模块 26，用于在分配和释放扇区链表时，判断相应的内部控制流项是否连续；

[0117] 批量操作模块 28，用于当判断结果为连续时，对该连续的内部控制流项进行批量操作。

[0118] 其中，内部控制流为主扇区分配表、扇区分配表、短流存放流、短扇区分配表及目录流中的一种。

[0119] 上述装置实施例中的具体概念和执行方式与前述的方法实施例中的一致，此处不做赘述。

[0120] 运用本发明实施例提出的预分配策略存储复合文档，并同时采取读缓存和批量写入的策略进行 I/O，可以大幅提升复合文档的打开、读写性能。同时，采用本发明实施例中的复合文档的存储方法获得的复合文档完全兼容现有的复合文档。

[0121] 与现有的微软复合文档相比，本发明实施例中的复合文档的读取的速度是微软复合文档的 4 倍左右，写入的速度新复合文档是微软复合文档的 100 倍左右。如图 12 ～ 14 可以看出新复合文档的性能要远远超过微软的复合文档。

[0122] 在图 12 ～ 14 中，示意了采用本发明实施例中的方法与采用现有技术的方法的耗时对比。其中，图 12 是用微软复合文档（现有技术中的一种复合文档）接口打开 db1 的耗

时和本发明实施例中的复合文档接口打开 db2 耗时的结果对比；图 13 是用微软复合文档接口模拟读取 db1 中 7 个好友的消息记录和本发明实施例中的复合文档接口模拟读取 db2 中 7 个好友消息记录的耗时对比；图 14 则是用微软复合文档接口模拟写入 7 个好友的消息记录和用本发明实施例中的复合文档接口模拟写入 7 个好友消息记录的耗时对比。

[0123] 其中，上述 3 次实验用到的复合文档 db 是模拟即时通讯工具的消息记录的生成过程生成的：用微软的复合文档接口产生一个 600M 字节的复合文档数据（称为 db1）和用本发明实施例中的复合文档接口产生一个 600M 字节的复合文档数据（称为 db2）。

[0124] 通过以上的实施方式的描述，本领域的技术人员可以清楚地了解到各实施方式可借助软件加必需的通用硬件平台的方式来实现，当然也可以通过硬件。基于这样的理解，上述技术方案本质上或者说对现有技术做出贡献的部分可以以软件产品的形式体现出来，该计算机软件产品可以存储在计算机可读存储介质中，如 ROM/RAM、磁碟、光盘等，包括若干指令用以使得一台计算机设备（可以是个人计算机，服务器，或者网络设备等）执行各个实施例或者实施例的某些部分所述的方法。

[0125] 以上所述的实施方式，并不构成对该技术方案保护范围的限定。任何在上述实施方式的精神和原则之内所作的修改、等同替换和改进等，均应包含在该技术方案的保护范围之内。

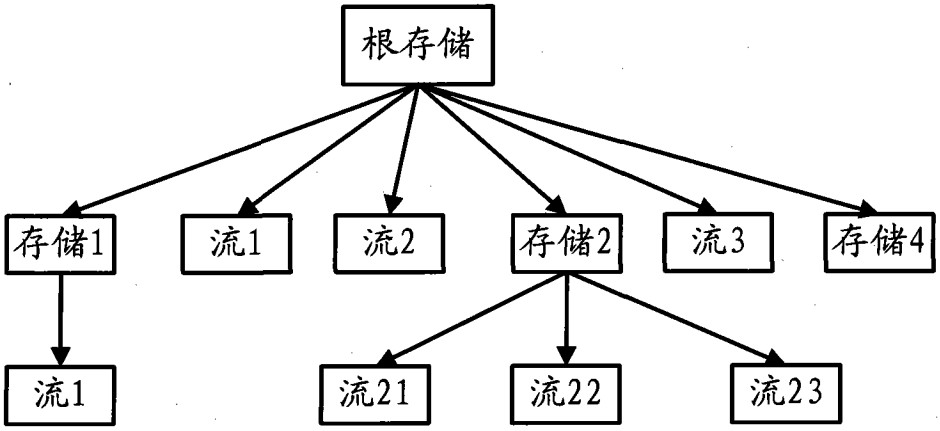

根存储
存储1
流1
流2
存储2
流3
存储4
流1
流21
流22
流23

图 1

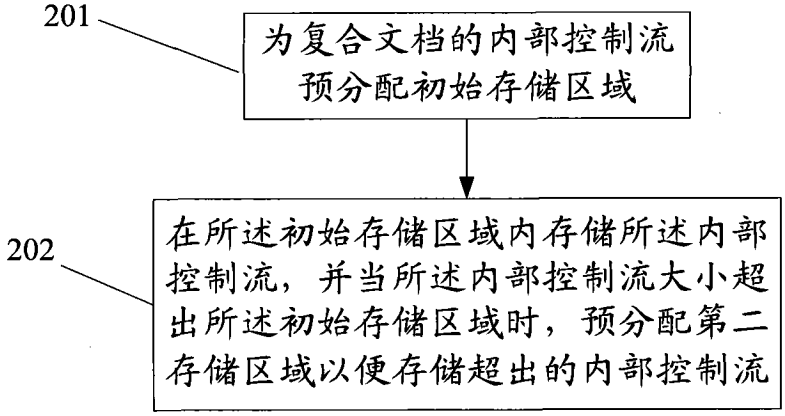

201
为复合文档的内部控制流预分配初始存储区域
202
在所述初始存储区域内存储所述内部控制流，并当所述内部控制流大小超出所述初始存储区域时，预分配第二存储区域以便存储超出的内部控制流

图 2

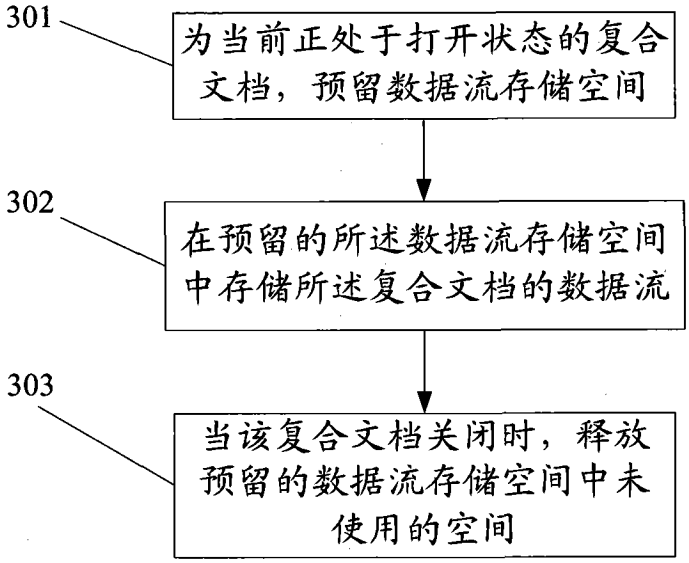

301
为当前正处于打开状态的复合文档，预留数据流存储空间
302
在预留的所述数据流存储空间中存储所述复合文档的数据流
303
当该复合文档关闭时，释放预留的数据流存储空间中未使用的空间

图 3

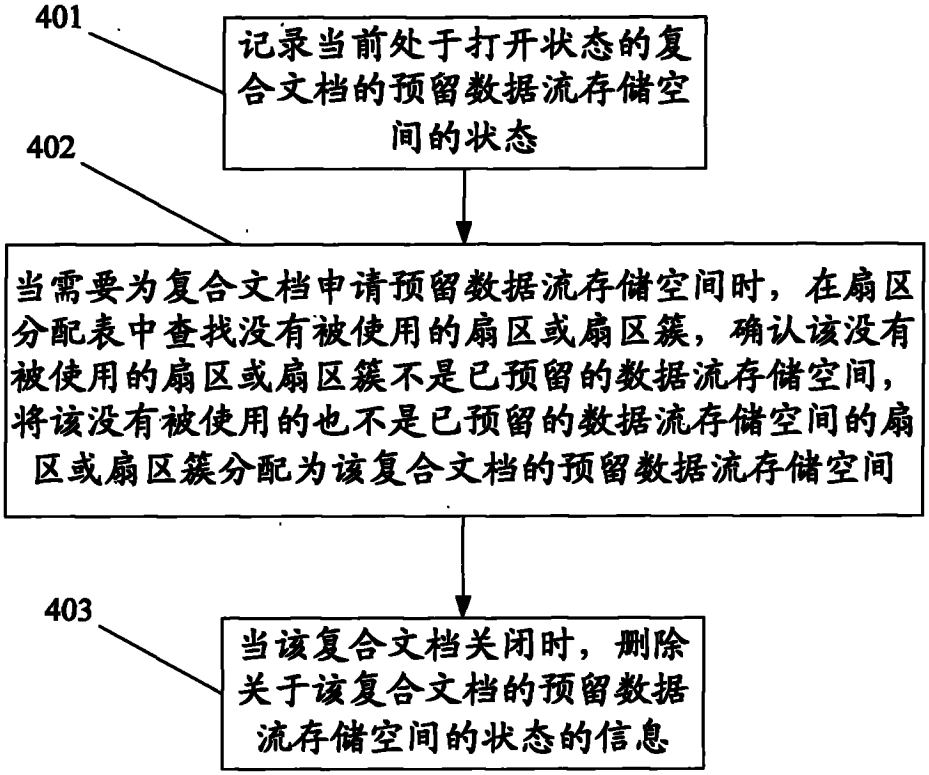

401
记录当前处于打开状态的复合文档的预留数据流存储空间的状态
402
当需要为复合文档申请预留数据流存储空间时，在扇区分配表中查找没有被使用的扇区或扇区簇，确认该没有被使用的扇区或扇区簇不是已预留的数据流存储空间，将该没有被使用的也不是已预留的数据流存储空间的扇区或扇区簇分配为该复合文档的预留数据流存储空间
403
当该复合文档关闭时，删除关于该复合文档的预留数据流存储空间的状态的信息

图 4

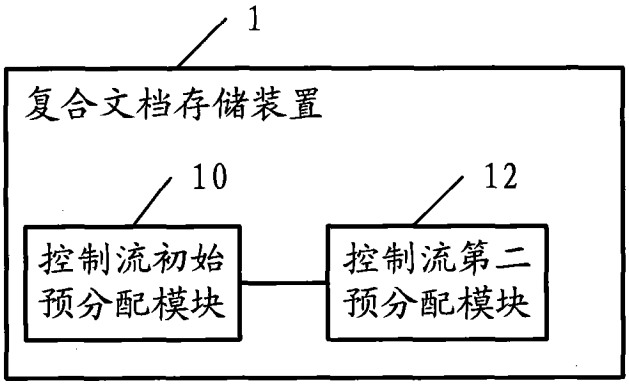

1
复合文档存储装置
10
控制流初始预分配模块
12
控制流第二预分配模块

图 5

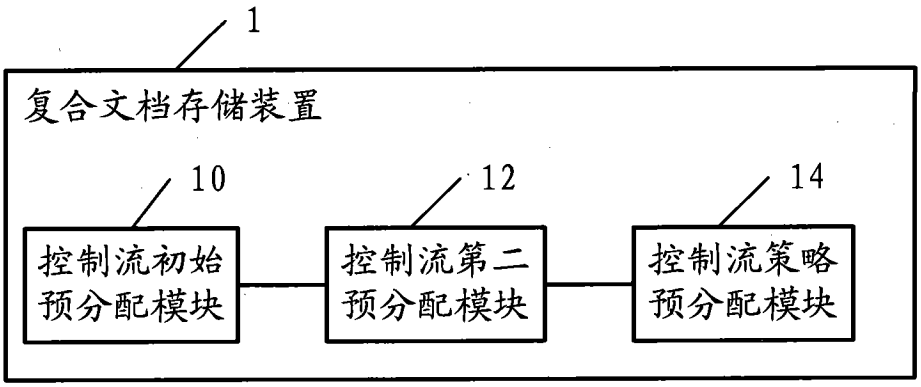

1
复合文档存储装置
10
控制流初始预分配模块
12
控制流第二预分配模块
14
控制流策略预分配模块

图 6

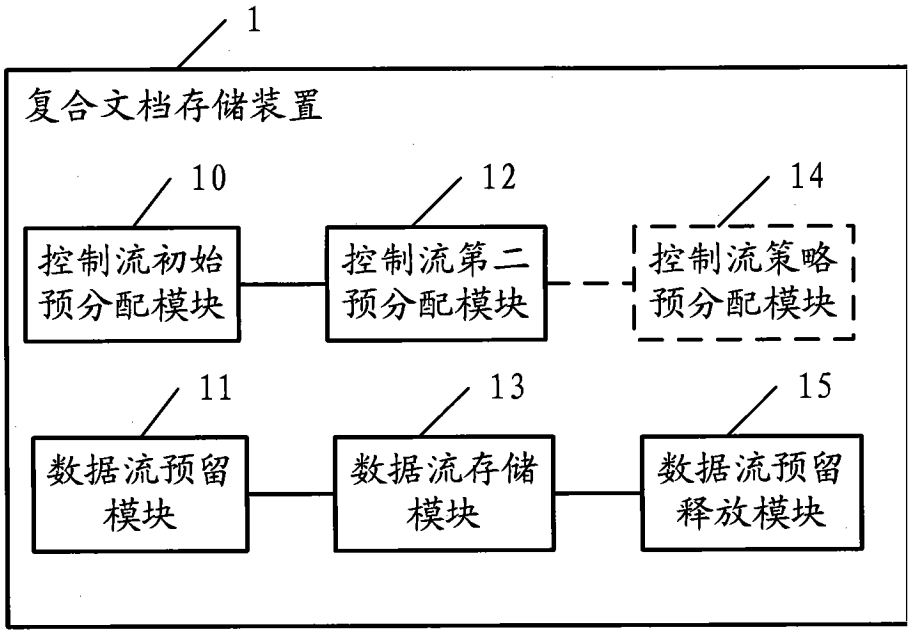

1
复合文档存储装置
10
控制流初始预分配模块
12
控制流第二预分配模块
14
控制流策略预分配模块
11
数据流预留模块
13
数据流存储模块
15
数据流预留释放模块

图 7

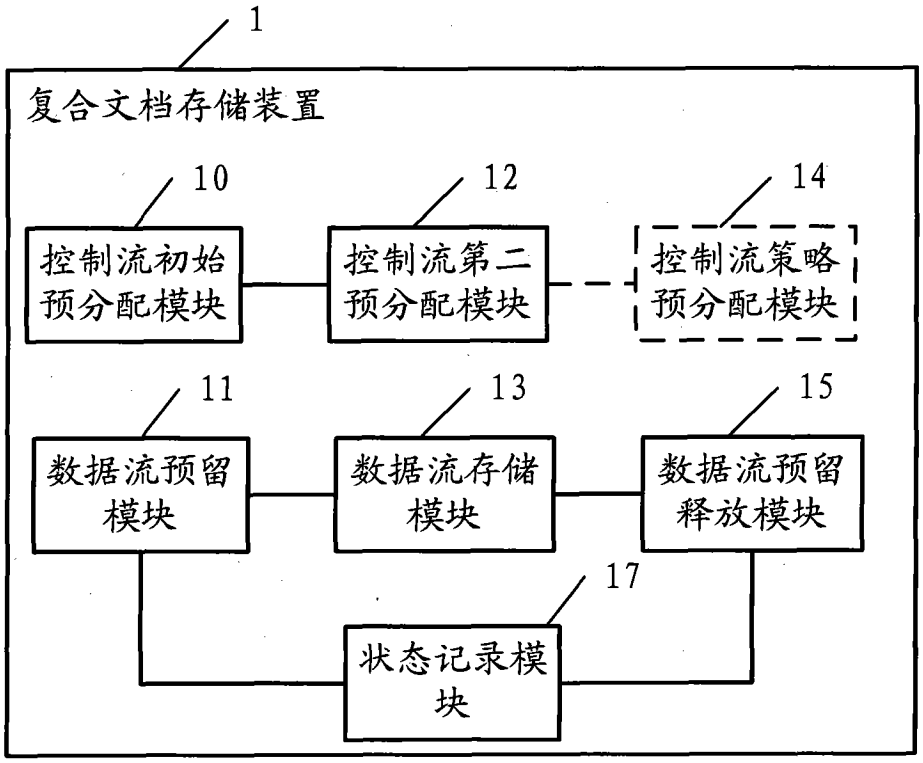

1
复合文档存储装置
10
控制流初始预分配模块
12
控制流第二预分配模块
14
控制流策略预分配模块
11
数据流预留模块
13
数据流存储模块
15
数据流预留释放模块
17
状态记录模块

图 8

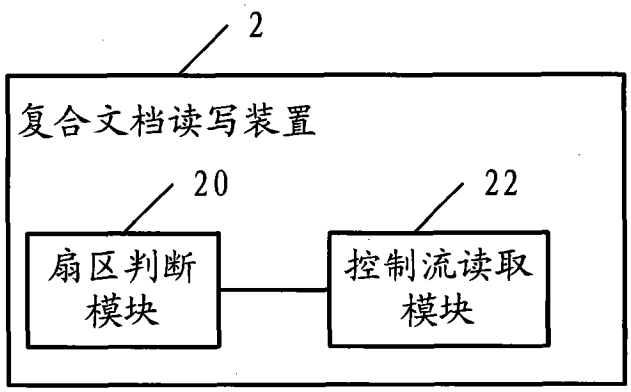

2
复合文档读写装置
20
扇区判断模块
22
控制流读取模块

图 9

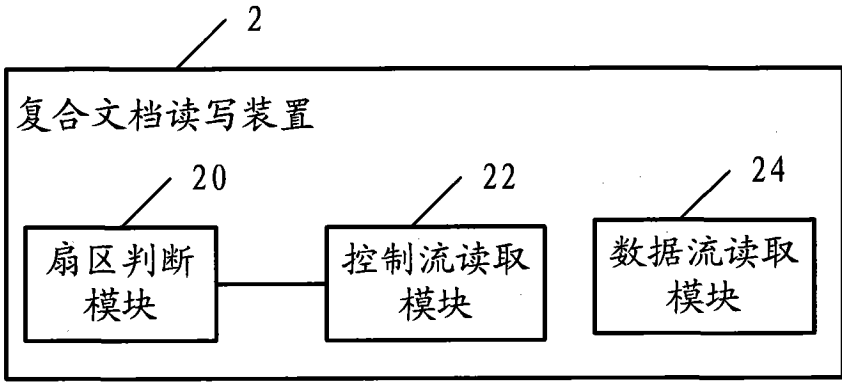

2
复合文档读写装置
20
扇区判断模块
22
控制流读取模块
24
数据流读取模块

图 10

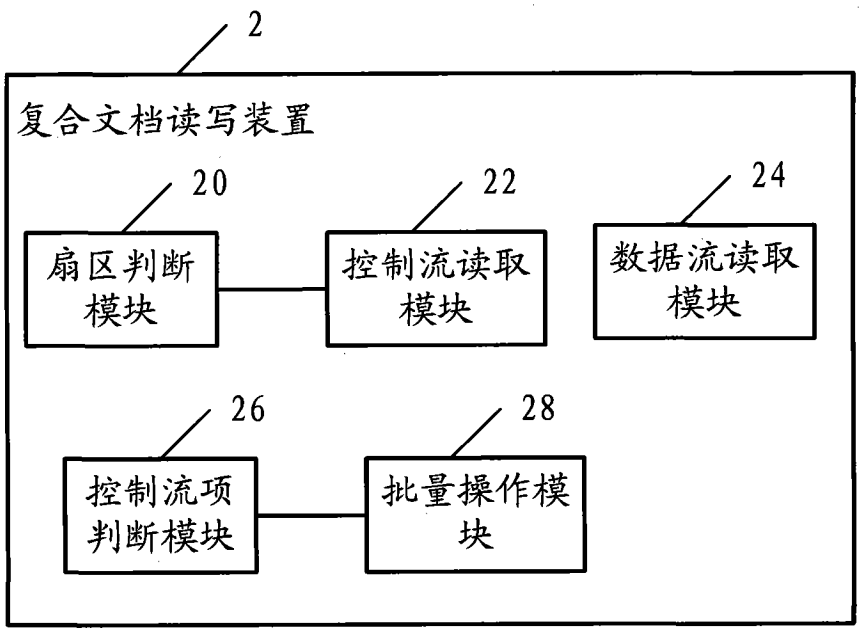

2
复合文档读写装置
20
扇区判断模块
22
控制流读取模块
24
数据流读取模块
26
控制流项判断模块
28
批量操作模块

图 11

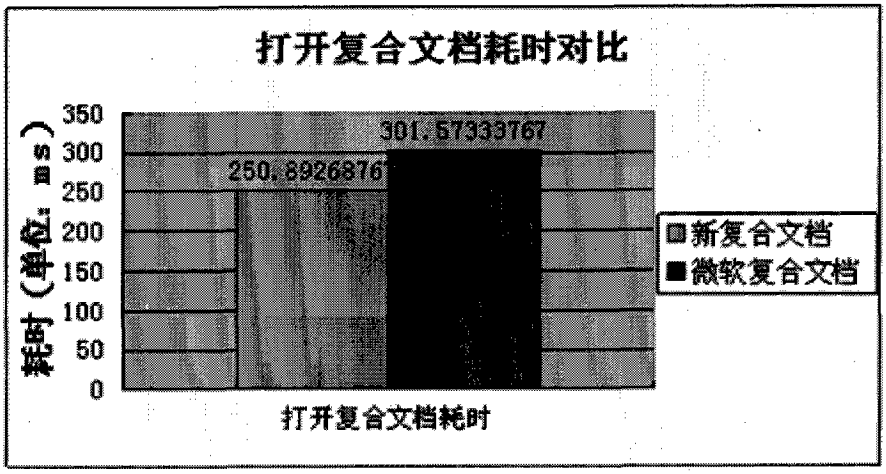

打开复合文档耗时对比
耗时（单位：ms）
350
300
250
200
150
100
50
0
301.57333767
250.8926876
新复合文档
微软复合文档
打开复合文档耗时

图 12

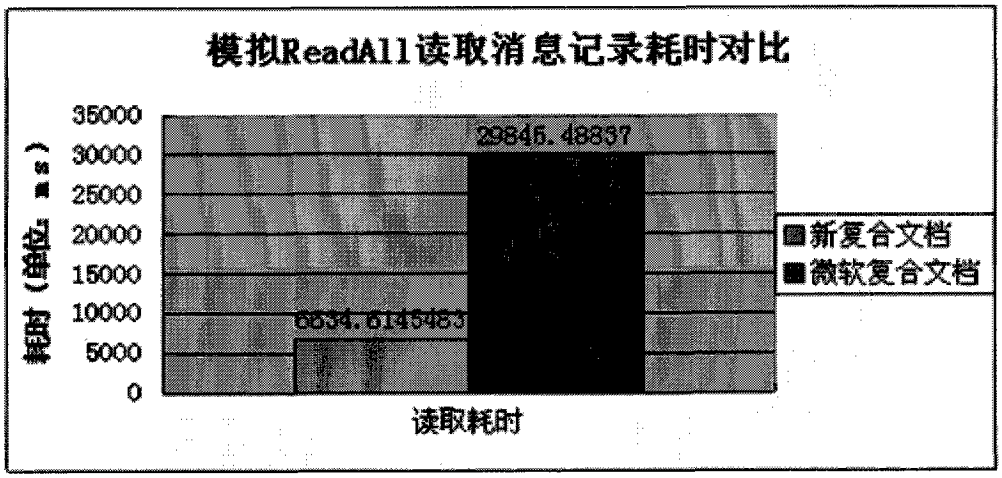

模拟ReadAll读取消息记录耗时对比
耗时（单位：ms）
35000
30000
25000
20000
15000
10000
5000
0
29846.48837
6634.6145483
新复合文档
微软复合文档
读取耗时

图 13

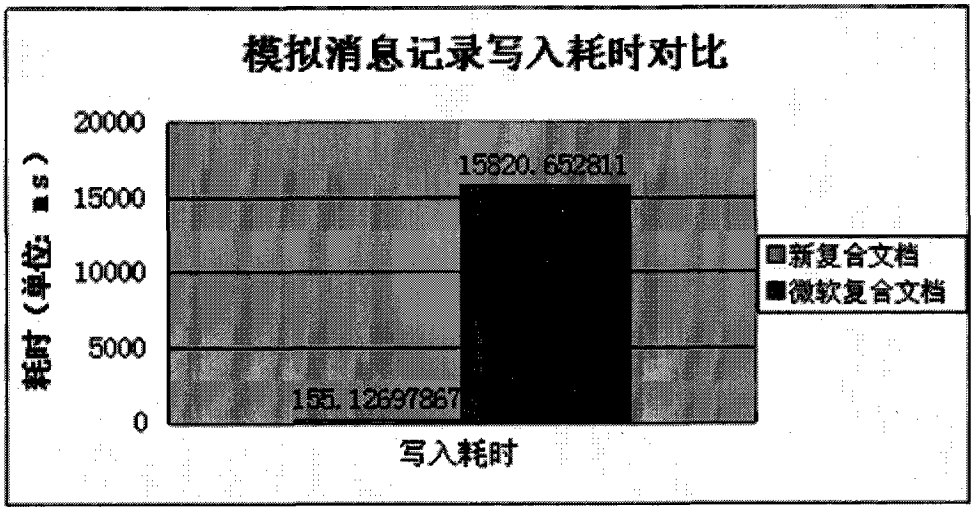

模拟消息记录写入耗时对比
20000
15000
10000
5000
0
耗时（单位：ms）
15820.652811
155.12697867
写入耗时
新复合文档
微软复合文档

图 14