



(12)发明专利

(10)授权公告号 CN 106537321 B

(45)授权公告日 2020.04.28

(21)申请号 201580014438.8

(72)发明人 施国军 李志刚 肖福洲

(22)申请日 2015.05.27

(74)专利代理机构 北京龙双利达知识产权代理有限公司 11329

(65)同一申请的已公布的文献号
申请公布号 CN 106537321 A

代理人 韩狄 毛威

(43)申请公布日 2017.03.22

(51)Int.Cl.

(85)PCT国际申请进入国家阶段日
2016.09.21

G06F 3/06(2006.01)

G06F 12/02(2006.01)

(86)PCT国际申请的申请数据
PCT/CN2015/079894 2015.05.27

(56)对比文件

US 2015089097 A1,2015.03.26,

JP H11317008 A,1999.11.16,

CN 103914522 A,2014.07.09,

(87)PCT国际申请的公布数据
W02016/187831 ZH 2016.12.01

审查员 姜晓盼

(73)专利权人 华为技术有限公司
地址 518129 广东省深圳市龙岗区坂田华为总部办公楼

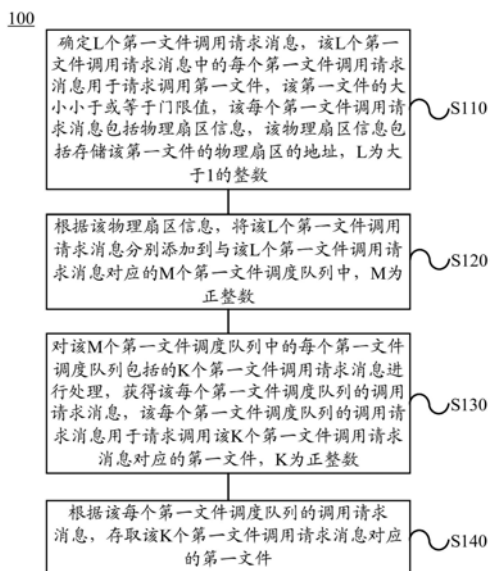
权利要求书4页 说明书18页 附图4页

(54)发明名称

存取文件的方法、装置和存储系统

(57)摘要

本发明实施例涉及存取文件的方法、装置和存储系统,该方法包括:确定L个第一文件调用请求消息;根据该L个第一文件调用请求消息中的物理扇区信息,将该L个第一文件调用请求消息分别添加到M个第一文件调度队列中;对每个第一文件调度队列包括的第一文件调用请求消息进行处理,获得每个第一文件调度队列的调用请求消息并用来存取第一文件。本发明实施例的存取文件的方法、装置和存储系统,通过为小文件的调用请求消息分配对应的调度队列,从而可以将多个小文件的调用请求消息进行合并处理,减少小文件的调用次数,可以避免小文件的调用请求而增加的I/O次数并导致的占用过多内存资源,并增加I/O请求的吞吐量,进而整体提升文件系统的I/O效率。



1. 一种存取文件的方法,其特征在于,包括:

确定L个用于请求调用文件大小小于或等于门限值的第一个文件的第一个文件调用请求消息,所述L个第一个文件调用请求消息中的每个第一个文件调用请求消息包括物理扇区信息,所述物理扇区信息包括存储所述第一个文件的物理扇区的地址,L为大于1的整数;

根据所述物理扇区信息,将所述L个第一个文件调用请求消息分别添加到与所述L个第一个文件调用请求消息对应的M个第一个文件调度队列中,M为正整数,其中,所述L个第一个文件调用请求消息中的至少两个第一个文件调用请求消息被添加到同一个第一个文件调度队列中,所述至少两个第一个文件调用请求消息请求调用的第一个文件的物理扇区的地址连续;

对所述M个第一个文件调度队列中的每个第一个文件调度队列包括的K个第一个文件调用请求消息进行处理,获得所述每个第一个文件调度队列的调用请求消息,所述每个第一个文件调度队列的调用请求消息用于请求调用所述K个第一个文件调用请求消息对应的第一个文件,K为正整数;

其中,所述对所述M个第一个文件调度队列中的每个第一个文件调度队列包括的K个第一个文件调用请求消息进行处理,包括:若所述每个第一个文件调度队列包括的至少两个第一个文件调用请求消息请求写入的第一个文件的大小之和恰好构成一个数据块,将所述至少两个第一个文件调用请求消息被合并为一个调用请求消息,使得所述至少两个第一个文件调用请求消息请求写入的第一个文件被写入同一个数据块的同一个物理扇区内;

根据所述每个第一个文件调度队列的调用请求消息,存取所述K个第一个文件调用请求消息对应的第一个文件。

2. 根据权利要求1所述的方法,其特征在于,所述方法还包括:

在存储文件的物理扇区上,确定N个第一个文件存储空间组,所述N个第一个文件存储空间组用于存储第一个文件,N为正整数;

当所述N个第一个文件存储空间组中至少两个第一个文件存储空间组存在可用存储空间时,将所述至少两个第一个文件存储空间组包括的第一个文件进行合并;

当所述N个第一个文件存储空间组中至少一个第一个文件存储空间组不存储文件时,释放所述至少一个第一个文件存储空间组;

当所述N个第一个文件的存储空间组都不存在所述可用存储空间时,增加所述N个第一个文件的存储空间组的个数;

所述根据所述每个第一个文件调度队列的调用请求消息,存取所述K个第一个文件调用请求消息对应的第一个文件,包括:

根据所述每个第一个文件调度队列的调用请求消息,从所述N个第一个文件存储空间组中存取所述K个第一个文件调用请求消息对应的第一个文件。

3. 根据权利要求1或2所述的方法,其特征在于,所述每个第一个文件调用请求消息还包括存取路径信息,所述存取路径信息用于指示存取所述每个第一个文件调用消息对应的第一个文件的物理扇区的路径。

4. 根据权利要求3所述的方法,其特征在于,所述确定L个用于请求调用文件大小小于或等于门限值的第一个文件的第一个文件调用请求消息,包括:

确定L个第一个文件信息节点,所述L个第一个文件信息节点中的每个第一个文件信息节点用于存储所述每个第一个文件调用请求消息请求调用的第一个文件的所述存取路径信息;

根据所述每个第一文件信息节点,确定所述每个第一文件调用请求消息请求调用的第一文件的所述存取路径信息。

5. 根据权利要求4所述的方法,其特征在于,所述确定L个第一文件信息节点,包括:

在信息节点链表中确定是否存在所述L个第一文件信息节点中第i个第一文件信息节点;

当所述信息节点链表中不存在所述第i个第一文件信息节点时,在所述信息节点链表中创建所述第i个第一文件信息节点,并将第i个第一文件的所述存取路径信息存储到所述第i个第一文件信息节点中;

当所述信息节点链表中存在所述第i个第一文件信息节点时,根据所述第i个第一文件信息节点确定所述第i个第一文件的所述存取路径信息。

6. 一种存取文件的装置,其特征在于,包括:

第一确定模块,用于确定L个用于请求调用文件大小小于或等于门限值的所述第一文件的第一文件调用请求消息,所述L个第一文件调用请求消息中的每个第一文件调用请求消息包括物理扇区信息,所述物理扇区信息包括存储所述第一文件的物理扇区的地址,L为大于1的整数;

分配模块,用于根据所述物理扇区信息,将所述第一确定模块确定的所述L个第一文件调用请求消息分别添加到与所述L个第一文件调用请求消息对应的M个第一文件调度队列中,M为正整数,其中,所述L个第一文件调用请求消息中的至少两个第一文件调用请求消息被添加到同一个第一文件调度队列中,所述至少两个第一文件调用请求消息请求调用的第一文件的物理扇区的地址连续;

处理模块,用于对所述分配模块分配的所述M个第一文件调度队列中的每个第一文件调度队列包括的K个第一文件调用请求消息进行处理,获得所述每个第一文件调度队列的调用请求消息,所述每个第一文件调度队列的调用请求消息用于请求调用所述K个第一文件调用请求消息对应的第一文件,K为正整数;

其中,所述处理模块具体用于,若所述每个第一文件调度队列包括的至少两个第一文件调用请求消息请求写入的第一文件的大小之和恰好构成一个数据块,将所述至少两个第一文件调用请求消息被合并为一个调用请求消息,使得所述至少两个第一文件调用请求消息请求写入的第一文件被写入同一个数据块的同一个物理扇区内;

调用模块,用于根据所述处理模块获得的所述每个第一文件调度队列的调用请求消息,存取所述K个第一文件调用请求消息对应的第一文件。

7. 根据权利要求6所述的装置,其特征在于,所述装置还包括:

第二确定模块,用于在存储文件的物理扇区上,确定N个第一文件存储空间组,所述N个第一文件存储空间组用于存储所述第一文件,N为正整数;

当所述N个第一文件存储空间组中至少两个第一文件存储空间组存在可用存储空间时,将所述至少两个第一文件存储空间组包括的第一文件进行合并,或

当所述N个第一文件存储空间组中至少一个第一文件存储空间组不存储文件时,释放所述至少一个第一文件存储空间组,或

当所述N个第一文件的存储空间组都不存在所述可用存储空间时,增加所述N个第一文件的存储空间组的个数;

所述调用模块还用于：

根据所述每个第一文件调度队列的调用请求消息，从所述N个第一文件存储空间组中存取所述K个第一文件调用请求消息对应的第一文件。

8. 根据权利要求6或7所述的装置，其特征在于，所述每个第一文件调用请求消息还包括存取路径信息，所述存取路径信息用于指示存取所述每个第一文件调用消息对应的第一文件的物理扇区的路径。

9. 根据权利要求8所述的装置，其特征在于，所述第一确定模块具体用于：

确定L个第一文件信息节点，所述L个第一文件信息节点中的每个第一文件信息节点用于存储所述每个第一文件调用请求消息请求调用的第一文件的所述存取路径信息；

根据所述每个第一文件信息节点，确定所述每个第一文件调用请求消息请求调用的第一文件的所述存取路径信息。

10. 根据权利要求9所述的装置，其特征在于，所述第一确定模块具体用于：

在信息节点链表中确定是否存在所述L个第一文件信息节点中第i个第一文件信息节点；

当所述信息节点链表中不存在所述第i个第一文件信息节点时，在所述信息节点链表中创建所述第i个第一文件信息节点，并将第i个第一文件的所述存取路径信息存储到所述第i个第一文件信息节点中；

当所述信息节点链表中存在所述第i个第一文件信息节点时，根据所述第i个第一文件信息节点确定所述第i个第一文件的所述存取路径信息。

11. 一种存储系统，其特征在于，包括：

存储器，用于存储文件和程序；

处理器，用于执行所述程序，以实现：

确定L个用于请求调用文件大小小于或等于门限值的第一文件的第一文件调用请求消息，所述L个第一文件调用请求消息中的每个第一文件调用请求消息包括物理扇区信息，所述物理扇区信息包括存储所述第一文件的物理扇区的地址，L为大于1的整数；

根据所述物理扇区信息，将所述L个第一文件调用请求消息分别添加到与所述L个第一文件调用请求消息对应的M个第一文件调度队列中，M为正整数，其中，所述L个第一文件调用请求消息中的至少两个第一文件调用请求消息被添加到同一个第一文件调度队列中，所述至少两个第一文件调用请求消息请求调用的第一文件的物理扇区的地址连续；

对所述M个第一文件调度队列中的每个第一文件调度队列包括的K个第一文件调用请求消息进行处理，获得所述每个第一文件调度队列的调用请求消息，所述每个第一文件调度队列的调用请求消息用于请求调用所述K个第一文件调用请求消息对应的第一文件，K为正整数；

其中，所述处理器具体用于，若所述每个第一文件调度队列包括的至少两个第一文件调用请求消息请求写入的第一文件的大小之和恰好构成一个数据块，将所述至少两个第一文件调用请求消息被合并为一个调用请求消息，使得所述至少两个第一文件调用请求消息请求写入的第一文件被写入同一个数据块的同一个物理扇区内；

根据所述每个第一文件调度队列的调用请求消息，存取所述K个第一文件调用请求消息对应的第一文件。

12. 根据权利要求11所述的存储系统,其特征在于,所述处理器还用于:

在存储文件的物理扇区上,确定N个第一文件存储空间组,所述N个第一文件存储空间组用于存储所述第一文件,N为正整数;

当所述N个第一文件存储空间组中至少两个第一文件存储空间组存在可用存储空间时,将所述至少两个第一文件存储空间组包括的第一文件进行合并,或

当所述N个第一文件存储空间组中至少一个第一文件存储空间组不存储文件时,释放所述至少一个第一文件存储空间组,或

当所述N个第一文件的存储空间组都不存在所述可用存储空间时,增加所述N个第一文件的存储空间组的个数;

根据所述每个第一文件调度队列的调用请求消息,从所述N个第一文件存储空间组中存取所述K个第一文件调用请求消息对应的第一文件。

13. 根据权利要求11或12所述的存储系统,其特征在于,所述每个第一文件调用请求消息还包括存取路径信息,所述存取路径信息用于指示存取所述每个第一文件调用消息对应的第一文件的物理扇区的路径。

14. 根据权利要求13所述的存储系统,其特征在于,所述处理器具体用于:

确定L个第一文件信息节点,所述L个第一文件信息节点中的每个第一文件信息节点用于存储所述每个第一文件调用请求消息请求调用的第一文件的所述存取路径信息;

根据所述每个第一文件信息节点,确定所述每个第一文件调用请求消息请求调用的第一文件的所述存取路径信息。

15. 根据权利要求14所述的存储系统,其特征在于,所述处理器具体用于:

在信息节点链表中确定是否存在所述L个第一文件信息节点中第i个第一文件信息节点;

当所述信息节点链表中不存在所述第i个第一文件信息节点时,在所述信息节点链表中创建所述第i个第一文件信息节点,并将第i个第一文件的所述存取路径信息存储到所述第i个第一文件信息节点中;

当所述信息节点链表中存在所述第i个第一文件信息节点时,根据所述第i个第一文件信息节点确定所述第i个第一文件的所述存取路径信息。

存取文件的方法、装置和存储系统

技术领域

[0001] 本发明涉及信息技术领域,尤其涉及存取文件的方法、装置和存储系统。

背景技术

[0002] 文件系统通常为每一个文件分配一个inode (information node,信息节点),为每个文件对应的实际数据分配数据块data block,并将data block的号码记录到inode中。文件系统中所支持的data block大小有1K,2K,4K等。在格式化时data block的大小就固定了,且每个data block都有编号,以方便inode的记录。

[0003] 由于现有技术中每个data block仅能容纳一个文件的数据,而文件系统中往往存在大量的小文件,比如几百个字节或者更小。如果为这样的小文件分配单独的data block,便会引起内部碎片,浪费磁盘空间。而且,频繁的小文件读写还会增加输入(Input,I)/输出(Output,O)次数,产生过多的系统调用,系统调用也会引起上下文切换,占用过多的CPU和内存资源,从而降低文件系统的整体I/O效率。

发明内容

[0004] 本发明提供了一种存取文件的方法、装置和存储系统,能够整体提升文件系统的I/O效率。

[0005] 第一方面,提供了一种存取文件的方法,该方法包括:确定L个第一文件调用请求消息,该L个第一文件调用请求消息中的每个第一文件调用请求消息用于请求调用第一文件,该第一文件的大小小于或等于门限值,该每个第一文件调用请求消息包括物理扇区信息,该物理扇区信息包括存储该第一文件的物理扇区的地址,L为大于1的整数;根据该物理扇区信息,将该L个第一文件调用请求消息分别添加到与该L个第一文件调用请求消息对应的M个第一文件调度队列中,M为正整数;对该M个第一文件调度队列中的每个第一文件调度队列包括的K个第一文件调用请求消息进行处理,获得该每个第一文件调度队列的调用请求消息,该每个第一文件调度队列的调用请求消息用于请求调用该K个第一文件调用请求消息对应的第一文件,K为正整数;根据该每个第一文件调度队列的调用请求消息,存取该K个第一文件调用请求消息对应的第一文件。

[0006] 结合第一方面,在第一方面的一种实现方式中,该方法还包括:在存储文件的物理扇区上,确定N个第一文件存储空间组,该N个第一文件存储空间组用于存储第一文件,N为正整数;当该N个第一文件存储空间组中至少两个第一文件存储空间组存在可用存储空间时,将该至少两个第一文件存储空间组包括的第一文件进行合并;当该N个第一文件存储空间组中至少一个第一文件存储空间组不存储文件时,释放该至少一个第一文件存储空间组;当该N个第一文件的存储空间组都不存在该可用存储空间时,增加该N个第一文件的存储空间组的个数;该根据该每个第一文件调度队列的调用请求消息,存取该K个第一文件调用请求消息对应的第一文件,包括:根据该每个第一文件调度队列的调用请求消息,从该N个第一文件存储空间组中存取该K个第一文件调用请求消息对应的第一文件。

[0007] 结合第一方面及其上述实现方式,在第一方面的另一种实现方式中,该根据该物理扇区信息,将该L个第一文件调用请求消息分别添加到与该L个第一文件调用请求消息对应的M个第一文件调度队列中,包括:根据该物理扇区信息,将该L个第一文件调用请求消息中的至少两个第一文件调用请求消息添加到同一个第一文件调度队列中,该至少两个第一文件调用请求消息请求调用的第一文件的物理扇区的地址连续。

[0008] 结合第一方面及其上述实现方式,在第一方面的另一种实现方式中,该每个第一文件调用请求消息还包括存取路径信息,该存取路径信息用于指示存取该每个第一文件调用消息对应的第一文件的物理扇区的路径。

[0009] 结合第一方面及其上述实现方式,在第一方面的另一种实现方式中,该确定L个第一文件调用请求消息,包括:确定L个第一文件信息节点,该L个第一文件信息节点中的每个第一文件信息节点用于存储该每个第一文件调用请求消息请求调用的第一文件的该存取路径信息;根据该每个第一文件信息节点,确定该每个第一文件调用请求消息请求调用的第一文件的该存取路径信息。

[0010] 结合第一方面及其上述实现方式,在第一方面的另一种实现方式中,该确定L个第一文件信息节点,包括:在信息节点链表中确定是否存在该L个第一文件信息节点中第i个第一文件信息节点;当该信息节点链表中不存在该第i个第一文件信息节点时,在该信息节点链表中创建该第i个第一文件信息节点,并将第i个第一文件的该存取路径信息存储到该第i个第一文件信息节点中;当该信息节点链表中存在该第i个第一文件信息节点时,根据该第i个第一文件信息节点确定该第i个第一文件的该存取路径信息。

[0011] 第二方面,提供了一种存取文件的装置,该装置包括:第一确定模块,用于确定L个第一文件调用请求消息,该L个第一文件调用请求消息中的每个第一文件调用请求消息用于请求调用第一文件,该第一文件的大小小于或等于门限值,该每个第一文件调用请求消息包括物理扇区信息,该物理扇区信息包括存储该第一文件的物理扇区的地址,L为大于1的整数;分配模块,用于根据该物理扇区信息,将该第一确定模块确定的该L个第一文件调用请求消息分别添加到与该L个第一文件调用请求消息对应的M个第一文件调度队列中,M为正整数;处理模块,用于对该分配模块分配的该M个第一文件调度队列中的每个第一文件调度队列包括的K个第一文件调用请求消息进行处理,获得该每个第一文件调度队列的调用请求消息,该每个第一文件调度队列的调用请求消息用于请求调用该K个第一文件调用请求消息对应的第一文件,K为正整数;调用模块,用于根据该处理模块获得的该每个第一文件调度队列的调用请求消息,存取该K个第一文件调用请求消息对应的第一文件。

[0012] 结合第二方面,在第二方面的一种实现方式中,该装置还包括:第二确定模块,用于在存储文件的物理扇区上,确定N个第一文件存储空间组,该N个第一文件存储空间组用于存储该第一文件,当该N个第一文件存储空间组中至少两个第一文件存储空间组存在可用存储空间时,将该至少两个第一文件存储空间组包括的第一文件进行合并,或当该N个第一文件存储空间组中至少一个第一文件存储空间组不存储文件时,释放该至少一个第一文件存储空间组,或当该N个第一文件的存储空间组都不存在该可用存储空间时,增加该N个第一文件的存储空间组的个数;该调用模块还用于:根据该每个第一文件调度队列的调用请求消息,从该N个第一文件存储空间组中存取该K个第一文件调用请求消息对应的第一文件。

[0013] 结合第二方面及其上述实现方式,在第二方面的另一种实现方式中,该处理模块具体用于:根据该物理扇区信息,将该L个第一文件调用请求消息中的至少两个第一文件调用请求消息添加到同一个第一文件调度队列中,该至少两个第一文件调用请求消息请求调用的第一文件的物理扇区的地址连续。

[0014] 结合第二方面及其上述实现方式,在第二方面的另一种实现方式中,该每个第一文件调用请求消息还包括存取路径信息,该存取路径信息用于指示存取该每个第一文件调用消息对应的第一文件的物理扇区的路径。

[0015] 结合第二方面及其上述实现方式,在第二方面的另一种实现方式中,该确定模块具体用于:确定L个第一文件信息节点,该L个第一文件信息节点中的每个第一文件信息节点用于存储该每个第一文件调用请求消息请求调用的第一文件的该存取路径信息;根据该每个第一文件信息节点,确定该每个第一文件调用请求消息请求调用的第一文件的该存取路径信息。

[0016] 结合第二方面及其上述实现方式,在第二方面的另一种实现方式中,该确定模块具体用于:在信息节点链表中确定是否存在该L个第一文件信息节点中第i个第一文件信息节点;当该信息节点链表中不存在该第i个第一文件信息节点时,在该信息节点链表中创建该第i个第一文件信息节点,并将第i个第一文件的该存取路径信息存储到该第i个第一文件信息节点中;当该信息节点链表中存在该第i个第一文件信息节点时,根据该第i个第一文件信息节点确定该第i个第一文件的该存取路径信息。

[0017] 第三方面,提供了一种存储系统,该存储系统包括:存储器,用于存储文件和程序;处理器,用于执行该程序,以实现:确定L个第一文件调用请求消息,该L个第一文件调用请求消息中的每个第一文件调用请求消息用于请求调用第一文件,该第一文件的大小小于或等于门限值,该每个第一文件调用请求消息包括物理扇区信息,该物理扇区信息包括存储该第一文件的物理扇区的地址,L为大于1的整数;根据该物理扇区信息,将该L个第一文件调用请求消息分别添加到与该L个第一文件调用请求消息对应的M个第一文件调度队列中,M为正整数;对该M个第一文件调度队列中的每个第一文件调度队列包括的K个第一文件调用请求消息进行处理,获得该每个第一文件调度队列的调用请求消息,该每个第一文件调度队列的调用请求消息用于请求调用该K个第一文件调用请求消息对应的第一文件,K为正整数;根据该每个第一文件调度队列的调用请求消息,存取该K个第一文件调用请求消息对应的第一文件。

[0018] 结合第三方面,在第三方面的一种实现方式中,该处理器还用于:该处理器还用于:在存储文件的物理扇区上,确定N个第一文件存储空间组,该N个第一文件存储空间组用于存储该第一文件,当该N个第一文件存储空间组中至少两个第一文件存储空间组存在可用存储空间时,将该至少两个第一文件存储空间组包括的第一文件进行合并,或当该N个第一文件存储空间组中至少一个第一文件存储空间组不存储文件时,释放该至少一个第一文件存储空间组,或当该N个第一文件的存储空间组都不存在该可用存储空间时,增加该N个第一文件的存储空间组的个数;根据该每个第一文件调度队列的调用请求消息,从该N个第一文件存储空间组中存取该K个第一文件调用请求消息对应的第一文件。

[0019] 结合第三方面及其上述实现方式,在第三方面的另一种实现方式中,该处理器具体用于:根据该物理扇区信息,将该L个第一文件调用请求消息中的至少两个第一文件调用

请求消息添加到同一个第一文件调度队列中,该至少两个第一文件调用请求消息请求调用的第一文件的物理扇区的地址连续。

[0020] 结合第三方面及其上述实现方式,在第三方面的另一种实现方式中,该每个第一文件调用请求消息还包括存取路径信息,该存取路径信息用于指示存取该每个第一文件调用消息对应的第一文件的物理扇区的路径。

[0021] 结合第三方面及其上述实现方式,在第三方面的另一种实现方式中,该处理器具体用于:确定L个第一文件信息节点,该L个第一文件信息节点中的每个第一文件信息节点用于存储该每个第一文件调用请求消息请求调用的第一文件的该存取路径信息;根据该每个第一文件信息节点,确定该每个第一文件调用请求消息请求调用的第一文件的该存取路径信息。

[0022] 结合第三方面及其上述实现方式,在第三方面的另一种实现方式中,该处理器具体用于:在信息节点链表中确定是否存在该L个第一文件信息节点中第i个第一文件信息节点;当该信息节点链表中不存在该第i个第一文件信息节点时,在该信息节点链表中创建该第i个第一文件信息节点,并将第i个第一文件的该存取路径信息存储到该第i个第一文件信息节点中;当该信息节点链表中存在该第i个第一文件信息节点时,根据该第i个第一文件信息节点确定该第i个第一文件的该存取路径信息。

[0023] 基于上述技术方案,本发明实施例的存取文件的方法、装置和存储系统,通过确定小文件的调用请求消息,并为这些小文件的调用请求消息分配对应的第一文件调度队列,从而可以将多个小文件的调用请求消息进行合并和/或删除处理,减少小文件的调用次数,可以避免小文件的调用请求而增加的I/O次数并导致的占用过多内存资源,可以减少系统调用次数,并增加I/O请求的吞吐量,进而整体提升文件系统的I/O效率。

附图说明

[0024] 为了更清楚地说明本发明实施例的技术方案,下面将对本发明实施例中所需要使用的附图作简单地介绍,显而易见地,下面所描述的附图仅仅是本发明的一些实施例,对于本领域普通技术人员来讲,在不付出创造性劳动的前提下,还可以根据这些附图获得其他的附图。

[0025] 图1是根据本发明实施例的存取文件的方法的示意性流程图。

[0026] 图2是根据本发明实施例的存取文件的方法的另一示意性流程图。

[0027] 图3是根据本发明实施例的存取文件的装置200的示意性框图。

[0028] 图4是根据本发明实施例的存取文件的装置200的另一示意性框图。

[0029] 图5是根据本发明实施例的存储系统300的示意性框图。

具体实施方式

[0030] 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,所描述的实施例是本发明的一部分实施例,而不是全部实施例。基于本发明中的实施例,本领域普通技术人员在没有做出创造性劳动的前提下所获得的所有其他实施例,都应属于本发明保护的范围。

[0031] 经初步统计,linux系统由数以万记的文件组成,甚至达到百万级。文件系统使用

目录结构组织存储数据,并将其原信息和实际数据关联起来。linux支持许多不同的文件系统,例如:ext4,F2FS,Btrfs,Vfat等。linux内核通过一个额外的软件层VFS将不同的底层文件系统关联起来。

[0032] 根据I/O访问的流程,我们可以将系统分为五大模块:用户层(User Space),文件系统层,块设备层,多媒体卡(Multimedia Card,简称“MMC”)驱动层,硬件。

[0033] 这里的User Space包含所有在用户空间处理I/O请求的应用程序及标准库。

[0034] 其中,文件系统层,块设备层,MMC驱动层运行在内核空间。

[0035] 硬件是指MMC主机控制器,主要依赖于JEDEC标准和各厂家的硬件实现,不作为本专利讨论的重点。

[0036] 图1示出了根据本发明实施例的存取文件的方法100的示意性流程图,该方法100可以由文件系统内的块设备层执行。如图1所示,该方法100包括:

[0037] S110,确定L个第一文件调用请求消息,该L个第一文件调用请求消息中的每个第一文件调用请求消息用于请求调用第一文件,该第一文件的大小小于或等于门限值,该每个第一文件调用请求消息包括物理扇区信息,该物理扇区信息包括存储该第一文件的物理扇区的地址,L为大于1的整数;

[0038] S120,根据该物理扇区信息,将该L个第一文件调用请求消息分别添加到与该L个第一文件调用请求消息对应的M个第一文件调度队列中,M为正整数;

[0039] S130,对该M个第一文件调度队列中的每个第一文件调度队列包括的K个第一文件调用请求消息进行处理,获得该每个第一文件调度队列的调用请求消息,该每个第一文件调度队列的调用请求消息用于请求调用该K个第一文件调用请求消息对应的第一文件,K为正整数;

[0040] S140,根据该每个第一文件调度队列的调用请求消息,存取该K个第一文件调用请求消息对应的第一文件。

[0041] 具体地,从用户设备发送的文件调用请求消息中,确定出L个第一文件调用请求消息,L为大于1的整数,该第一文件调用请求消息用于调用第一文件,该第一文件即为小文件,每个第一文件调用请求消息中都包括物理扇区信息,该物理扇区信息包括存储对应的小文件的物理扇区的地址。为L个第一文件调用请求消息分配M个第一文件调度队列,M为正整数,并根据每个第一文件调用请求消息中的物理扇区信息,将该L个第一文件调用请求消息分别添加到M个第一文件调度队列中,将每个第一文件调度队列中被分配到的K个第一文件调用请求消息进行处理,例如可以进行合并和/或删除处理,从而获得每个第一文件调度队列对应的调用请求消息,根据每个第一文件调度队列的调用请求消息,对K个第一文件调用请求消息所对应的第一文件进行调用,例如可以进行存取该第一文件。

[0042] 因此,本发明实施例的存取文件的方法,通过确定小文件的调用请求消息,并为这些小文件的调用请求消息分配对应的第一文件调度队列,从而可以将多个小文件的调用请求消息进行合并和/或删除处理,减少小文件的调用次数,可以避免小文件的调用请求而增加的I/O次数并导致的占用过多内存资源,可以减少系统调用次数,并增加I/O请求的吞吐量,进而整体提升文件系统的I/O效率。

[0043] 在S110中,文件系统可以从接收到的文件调用请求消息中确定L个第一文件调用请求消息,L为大于1的正整数。该L个第一文件调用请求消息中每个第一文件请求消息用于

请求调用对应的一个第一文件。可选地,该第一文件为小文件(Micro File,简称“MF”),根据实际应用,可以设置门限值,将文件的大小小于或等于该门限值的文件确定为小文件,即第一文件,例如,门限值可以设置为1K,即将文件大小小于或等于1K的文件限定为小文件,门限值也可以设置为2M、4M等,本发明并不限于此,则文件系统可以从接收到的文件调用请求消息中确定小文件的调用请求消息,这些小文件的调用请求消息用于向文件系统请求调用对应的小文件。

[0044] 在本发明实施例中,文件系统确定L个第一文件调用请求消息,可以一次接受一个第一文件调用请求消息,也可以一次接收多个第一文件调用请求消息,例如,文件系统可以一次接收3个第一文件调用请求消息,则文件系统确定L个第一文件调用请求消息,即确定接收到的3个第一文件调用请求消息。也可以接收随机的第一文件调用请求消息,当积累到L个时,即确定L个第一文件调用请求消息,该L的值可以根据实际应用确定,也可以根据经验值设置,本发明并不限于此。

[0045] 可选地,在本发明实施例中,每个第一文件调用请求消息中可以包括物理扇区信息,该物理扇区信息包括存储该每个第一文件调用请求消息请求调用的第一文件的物理扇区的地址。可选地,该物理扇区信息中具体可以包括请求调用的第一文件的起始地址、被调用的第一文件的大小、以及下发该第一文件调用请求消息的设备的相关信息中的至少一个。文件系统可以根据包括该物理扇区信息的第一文件调用请求消息确定要调用的第一文件。

[0046] 可选地,在本发明实施例中,该第一文件调用请求消息中还可以包括存取路径信息,该存取路径信息包括对应的存储该第一文件的物理扇区的路径信息,根据该存取路径信息可以在文件系统中找到存储对应的第一文件的物理扇区,即存储该小文件的物理扇区。具体地,当打开open或关闭close一个文件或创建一个新文件时,文件系统的内核会分配一个文件描述符并返回给用户程序,该文件描述符表项中的指针指向新打开的文件。在存取文件时,用户程序把文件描述符传给内核,内核根据文件描述符找到相应的表项,再通过表项中的指针找到相应的文件。每个文件结构体都有一个指向dentry(directory entry,目录项)结构体的指针,我们可以根据路径找到文件的inode节点。为了减少读盘次数,内核一般会缓存了目录的树状结构,称为dentry cache,其中每个inode节点都是一个dentry结构体,只要沿着路径各部分的dentry搜索即可,dentry cache只保存最近访问过的目录项,如果要没有命中,就要从磁盘读到内存中。

[0047] 具体地,对于小文件,文件系统可以将小文件的信息节点MF-inode以链表的形式进行存储,这些小文件的信息节点分别存储了对应的小文件的存取路径信息,可以根据信息节点中的存取路径信息查找小文件。当文件系统接收到第i个第一文件调用请求消息时,在存储的MF-inode链表中查找对应的第i个第一文件的信息节点,若该链表中存在该第i个第一文件的信息节点,则根据该第i个第一文件的信息节点,确定用于指示该第一文件的存储路径的存取路径信息,可以将该存取路径信息存储到该第i个第一文件调用请求消息中。当在MF-inode链表中无法查找到对应的第i个第一文件的信息节点,则申请该第i个第一文件对应的新的inode,并将该新的inode加入到MF-inode链表中,同时将该第一文件对应的存储路径和文件名保存到该新的inode中,记录为该inode的成员变量,也将该存取路径信息存储到该第i个第一文件调用请求消息中。

[0048] 可选地,对于小文件对应的MF-inode链表,当其中包括的inode对应的文件由于文件数据的不断写入,从小文件变为大文件时,例如当该小文件的大小超过4M时,可以通过调用截断(O_TRUNC)操作,使得大文件被截断为小文件;也可以将变大的文件移除小文件的MF-inode链表中,为其分配普通inode,由于性能上没有什么影响,也可以保持现状。相反地,当大文件变为小文件时,可以将该小文件添加到MF-inode链表,可选地,可以通过磁盘碎片整理程序(defragment daemon)在系统空闲的时候完成。

[0049] 在本发明实施例中,Linux的I/O调度程序是块设备I/O子系统的主要组件,它介于通用块层和块设备驱动程序之间,当Linux内核组件要读写一些数据时,并不是请求一发出,内核便立即执行该请求,而是将其推迟执行。延迟分配是块设备性能的关键机制,当传输一个新数据块时,内核检查能否通过扩展前一个一直处于等待状态的请求而满足新请求。

[0050] 具体地,每个块设备程序都有一个请求队列与之关联。在块设备初始化时,会分配并初始化请求队列。在这个时候,我们便可以为块设备驱动程序指定特定的I/O调度算法,默认情况下是使用系统默认的调度算法。默认的调度算法会优先处理与上一个处理请求最近的请求,因而就会对某个请求忽略很长一段时间,即为电梯算法。为了避免这种情况,当请求被传给电梯算法是,就会启动超时定时器,从而来避免某个请求永远得不到处理的情况。为了提升用户体验,读请求将一般优先于写请求,因为读请求通常阻塞发出请求的进程。

[0051] 在本发明实施例中,可以采用这种算法对小文件的调用请求消息进行处理,可选地,还可以通过下面的I/O调度(scheduler)算法方法进行调度。在S120中,可以根据L个第一文件调用请求消息中的物理扇区信息,将该L个第一文件调用请求消息分别添加到M个第一文件调度队列(MF dispatch Queue)中,M为整数。文件系统为小文件的调用请求消息分配了对应的第一文件调度队列,对于文件系统接收到的第一文件调用请求消息,可以根据每个调用请求消息中的物理扇区信息,将L个第一文件调用请求消息分别添加到M个第一文件调度队列中,使得具有物理扇区地址连续的至少两个第一文件调用请求消息被添加到同一个第一文件调度队列中,其中,每个第一文件调度队列包括K个第一文件调用请求消息,K和M均为正整数。

[0052] 例如,在本发明实施例中,文件系统确定了6个第一文件调用请求消息,即 $L=6$,当只有一个第一文件调度队列,即小文件的调度队列时, $M=1$,而6个第一文件调用请求消息中存在三个第一文件调用请求消息中的物理扇区信息显示其物理扇区地址连续,另外三个不连续。将6个第一文件调用请求消息均放入该第一文件调度队列中,并将物理扇区地址连续三个第一文件调用请求消息进行合并,即此时 $K=3$;而另外三个物理扇区地址不连续的第一文件调用请求消息暂时不做处理,待有新的第一文件调用请求消息时,再进行物理扇区地址是否连续的判断,若连续,则进行合并,如果不连续,则仍然不做处理。可选地,可以通过物理扇区地址是否连续来判断是否进行合并处理,还可以根据调用的第一文件的大小进行第一文件调用请求信息的合并,例如,当两个第一文件调用请求消息请求写入的两个第一文件的大小之和,恰好可以构成一个数据块(block),则也可以将这两个第一文件调用请求消息合并,并将对应的要求写入的第一文件写入同一个物理扇区内,同一个数据块中,本发明并不限于此。

[0053] 可选地,还例如,文件系统确定了6个第一文件调用请求消息,即 $L=6$,且将第一文件调度队列分为5个,即小文件的调度队列有5个, $M=5$ 时,而6个第一文件调用请求消息中存在三个第一文件调用请求消息中的物理扇区信息显示其物理扇区地址连续,另外三个不连续。则将该三个物理扇区地址连续的第一文件调用请求消息放入同一个第一文件调度队列中,例如将该三个第一文件调用请求消息放入第一个第一文件调度队列中,另外三个互相之间不连续的第一文件调用请求消息则可以放入另外三个第一文件调度队列中,而还有一个第一文件调度队列暂时为空。此时,第一个第一文件调度队列中存在3个第一文件调用请求消息,即 $K=3$,则可以对该第一文件调度队列中的三个第一文件调用请求消息进行合并处理,合并为一个第一文件调度队列的调用请求消息。而另外各有一个第一文件调用请求消息的三个第一调度队列则可以先不进行合并处理,继续等待分配其它第一文件调度请求消息,当这几个第一文件调度队列中存在多个可以合并的第一文件调用请求消息时,则同样地,与第一个第一文件调度队列一样,将在同一个第一文件调度队列中的多个第一文件调用请求消息进行合并。可选地,可以通过物理扇区地址是否连续来判断是否进行合并处理,还可以根据调用的第一文件的大小进行第一文件调用请求信息的合并,例如,当两个第一文件调用请求消息请求写入的两个第一文件的大小之和,恰好可以构成一个数据块(block),则也可以将这两个第一文件调用请求消息合并,并将对应的要求写入的第一文件写入同一个物理扇区内,同一个数据块中,本发明并不限于此。

[0054] 可选地,又例如,文件系统确定了6个第一文件调用请求消息,即 $L=6$,如果存在3个第一文件调度队列,即 $M=3$,并且6个第一文件调用请求消息中三个第一文件调用请求消息中的物理扇区信息显示其物理扇区地址连续,另外三个不连续,则先将该三个物理扇区地址连续的第一文件调用请求消息放入同一个第一文件调度队列中,例如将该三个第一文件调用请求消息放入第一个第一文件调度队列中,而对于另外三个物理地址不连续的第一文件调用请求消息,可以将他们按照大小分配或随机分配到三个第一文件调度队列中;或者也可以根据他们请求调用的第一文件的存储地址的相关性,将他们分配到三个第一文件调度队列中;还可以将他们随机分配到三个第一文件调度队列中。然后对第一个第一文件调度队列中物理扇区信息连续的三个第一文件调用请求消息进行合并处理,合并为一个第一文件调度队列的调用请求消息。而另外的两个第一文件调度队列可以先不进行处理,或者第一个第一文件调度队列中如果还有没有被合并的第一文件调用请求消息,也暂时不进行处理,当这几个第一文件调度队列中存在多个可以合并的第一文件调用请求消息时,则同样地,与第一个第一文件调度队列一样,将在同一个第一文件调度队列中的多个第一文件调用请求消息进行合并。可选地,可以通过物理扇区地址是否连续来判断是否进行合并处理,还可以根据调用的第一文件的大小进行第一文件调用请求信息的合并,例如,当两个第一文件调用请求消息请求写入的两个第一文件的大小之和,恰好可以构成一个数据块(block),则也可以将这两个第一文件调用请求消息合并,并将对应的要求写入的第一文件写入同一个物理扇区内,同一个数据块中,本发明并不限于此。

[0055] 可选地,可以按照 L 个第一文件调用请求消息的进程号(PID)的不同,将文件系统接收的调用请求消息进行分组。再将分组后的调用请求消息中的第一文件调用请求消息存储到第一文件调度队列,该第一文件调度队列可以包括 M 个调度队列。可选地,可以在文件系统中新增一个结构体,用于判断哪些是第一文件调用请求消息;也可以在S110中确定第

一文件调用请求消息时,将第一文件调用请求消息进行标识,通过该标识区分第一文件调用请求消息,并将第一文件调用请求消息分配到对应的M个第一文件调度队列中,每个第一文件调度对应可以包括K个第一文件调用请求消息。

[0056] 在S130中,文件系统中的M个第一文件调度队列中的每个第一文件调度队列可以对其包括的K个第一文件调用请求消息进行处理,获得该每个第一文件调度队列的调用请求消息。具体地,将L个第一文件调用请求消息添加至M个第一文件调度队列,对于每个第一文件调度队列,当其中包括的至少两个第一文件调用请求消息,具有连续的物理扇区 (bi_sector) 地址,可以将该至少两个第一文件调用请求消息进行合并处理,使得该至少两个第一文件调用请求消息可以合成为一个调用请求消息,即第一文件调度队列的调用请求消息,从而减少了调用的次数。例如,对于任意一个第一文件调度队列,该第一文件调度队列包括K个第一文件调用请求消息,根据该K个第一文件调用请求消息中包括的物理扇区信息,可以对该K个第一文件调用请求消息进行处理,生成一个该第一文件调度队列的调用请求消息,该第一文件调度队列的调用请求消息用于调用K个第一文件调用请求消息对应的第一文件。可选地,文件系统还可以对第一文件调用请求消息进行删除处理,删除不必要的调用请求,也可以减少调用次数。

[0057] 可选地,当某个第一文件调度队列只包括一个第一文件调用请求消息时,即不存在与该调用请求消息中物理扇区地址相连续的另一个调用请求消息时,可以暂时对该第一文件调用请求消息不作处理,当该第一文件调度队列中包括至少两个第一文件调用请求消息的物理扇区地址连续时,再进行合并处理,本发明并不限于此。

[0058] 在S140中,文件系统将第一文件调度队列中包括的K个第一文件调用请求消息进行合并或删除处理后,可以根据获得的第一文件调度队列的调用请求消息,存取相应的第一文件。具体地,由于每个第一文件调用请求消息中可以包括存取路径信息,在第一文件调度队列进行合并后,每个第一文件调度队列的调用请求消息仍然具有对应的存取路径信息,因此可以根据该存取路径信息,对应的对第一文件进行存取处理。可选地,第一文件调度队列的调用请求消息可以发送至存储介质请求队列 (mmc_request) 中,mmc_request处理第一文件调度队列的调用请求消息,并设置设备控制器,主要用于产生数据并传送,当数据传输完成后可以从对应的调度队列中删除当前的调用请求消息,并开始下一个请求对应的数据传输。

[0059] 应理解,在本发明的各种实施例中,上述各过程的序号的大小并不意味着执行顺序的先后,各过程的执行顺序应以其功能和内在逻辑确定,而不应对本发明实施例的实施过程构成任何限定。

[0060] 因此,本发明实施例的存取文件的方法,通过确定小文件的调用请求消息,并为这些小文件的调用请求消息分配对应的第一文件调度队列,从而可以将多个小文件的调用请求消息进行合并和/或删除处理,减少小文件的调用次数,可以避免小文件的调用请求而增加的I/O次数并导致的占用过多内存资源,可以减少系统调用次数,并增加I/O请求的吞吐量,进而整体提升文件系统的I/O效率。

[0061] 可选地,作为一个实施例,根据本发明实施例的存取文件的方法100还包括:

[0062] S150,在存储文件的物理扇区上,确定N个第一文件存储空间组,该N个第一文件存储空间组用于存储第一文件。

[0063] 在本发明实施例中,具体地,文件系统可以为小文件的存取设置单独的物理扇区,该物理扇区中包括多个单独的区域,该区域包括多个第一文件存储空间组(MF Group),该区域的大小可以根据时隙应用进行调整及优化,该区域内包括的MF Groups的个数可以根据需求动态调整。可选地,对于单独存放小文件的多个MF Group,每个MF Group可以包括多个数据库(data block),每个data block可以存储一个或多个小文件,本发明并不限于此。

[0064] 具体地,可以通过磁盘碎片整理程序(defragmentation daemon)在系统空闲时完成多个MF Group的合并和/或删除。例如,可以将多个MF Group进行合并,保证多个第一文件存储在连续的物理扇区上,进而可以更好地利用eMMC的特性。在系统的运行的过程中,由于文件的删除操作,使得某些MF Group中用于存储文件的空间有剩余,即该MF Group中大部分空间被加入了待删除的列表中,可以将这些MF Group进行合并,将至少两个MF Group中的数据合并存储到一个或多个MF Group中,从而释放出更多的空间,提升空间的利用率。

[0065] 可选地,作为一个实施例,也可以对MF Groups进行删除处理。例如,当MF Group中存在某些小文件变成大文件时,例如超过了4M的范围,可以找到该变大的文件所在的MF Group进行标记,然后当defragmentation daemon遍历待删除的列表,发现该MF Group中包括的第一文件均被加入到删除列表,使得该MF Group中已经不存在文件时,即该MF Groups为空时,会进行删除,删除该MF Group来释放资源。

[0066] 可选地,作为一个实施例,也可以对MF Group进行增加处理。例如,当新增加了小文件时,例如:某个大文件变成小文件时,defragmentation daemon首先会在多个MF Group中查找存在剩余空间的MF Group,如果存在可用存储空间可以将该小文件存到该MF Group中,则将新增加的小文件存储到该MF Group中;如果不存在有剩余空间的MF Groups,defragmentation daemon会增加新的MF Group,用来存放该小文件以及其它新增加的小文件。

[0067] 在本发明实施例,文件系统可以该根据该至少一个第一文件调度队列的调用请求消息,在存储文件的MF Group中查找需要存取调用的第一文件。

[0068] 应理解,在本发明的各种实施例中,上述各过程的序号的大小并不意味着执行顺序的先后,各过程的执行顺序应以其功能和内在逻辑确定,而不对本发明实施例的实施过程构成任何限定。

[0069] 因此,本发明实施例的存取文件的方法,通过确定小文件的调用请求消息,并为这些小文件的调用请求消息分配对应的第一文件调度队列,从而可以将多个小文件的调用请求消息进行合并和/或删除处理,可以避免小文件的调用请求而增加的I/O次数并导致的占用过多内存资源,可以减少系统调用次数,并增加I/O请求的吞吐量,进而整体提升文件系统的I/O效率。另外,还可以为小文件分配相应的动态存储区域,将不同的小文件聚合在一起存放,可以减少存储空间的浪费,提升空间的利用率。

[0070] 上文中结合图1和图2,详细描述了根据本发明实施例的存取文件的方法,下面将结合图3和图4,描述根据本发明实施例的存取文件的装置200。

[0071] 如图3所示,根据本发明实施例的存取文件的装置200包括:

[0072] 第一确定模块210,用于确定L个第一文件调用请求消息,该L个第一文件调用请求消息中的每个第一文件调用请求消息用于请求调用第一文件,该第一文件的大小小于或等于门限值,该每个第一文件调用请求消息包括物理扇区信息,该物理扇区信息包括存储该第一文件的物理扇区的地址,L为大于1的整数;

[0073] 分配模块220,用于根据该物理扇区信息,将该第一确定模块确定的该L个第一文件调用请求消息分别添加到与该L个第一文件调用请求消息对应的M个第一文件调度队列中,M为正整数;

[0074] 处理模块230,用于对该分配模块分配的该M个第一文件调度队列中的每个第一文件调度队列包括的K个第一文件调用请求消息进行处理,获得该每个第一文件调度队列的调用请求消息,该每个第一文件调度队列的调用请求消息用于请求调用该K个第一文件调用请求消息对应的第一文件,K为正整数;

[0075] 调用模块240,用于根据该处理模块获得的该每个第一文件调度队列的调用请求消息,存取该K个第一文件调用请求消息对应的第一文件。

[0076] 具体地,确定模块210从用户设备发送文件的调用请求消息中,确定出L个第一文件的调用请求消息,L为大于1的整数,该第一文件调用请求消息用于调用第一文件,该第一文件即为小文件,每个小文件的调用请求消息可以包括物理扇区信息,该物理扇区信息包括存储该小文件的物理扇区的地址。分配模块220为确定模块210确定的L个第一文件调用请求消息分配M个第一文件调度队列,M为正整数,并根据每个第一文件调用请求消息中的物理扇区信息,将该L个第一文件的调用请求消息分别添加到M个第一文件调度队列中,处理模块230针对每个第一文件调度队列,将被分配到其中的K个第一文件调用请求消息进行处理,例如可以进行合并和/或删除处理,从而获得每个第一文件调度队列对应的调用请求消息,调用模块240可以根据每个第一文件调度队列的调度请求消息,对K个第一文件调用请求消息所对应的第一文件进行调用,例如可以进行存取该第一文件。

[0077] 因此,本发明实施例的存取文件的装置,通过确定小文件的调用请求消息,并为这些小文件的调用请求消息分配对应的第一文件调度队列,从而可以将多个小文件的调用请求消息进行合并和/或删除处理,减少小文件的调用次数,可以避免小文件的调用请求而增加的I/O次数并导致的占用过多内存资源,可以减少系统调用次数,并增加I/O请求的吞吐量,进而整体提升文件系统的I/O效率。

[0078] 在本发明实施例中,文件系统存取文件的装置200的确定模块210可以从接收到的文件调用请求消息中确定L个第一文件调用请求消息,该L个第一文件调用请求消息中每个第一文件请求消息用于请求调用对应的一个第一文件。可选地,该第一文件为小文件(Micro File,简称“MF”),根据实际应用,可以设置门限值,将文件的大小小于或等于该门限值的文件确定为小文件,即第一文件,例如,门限值可以设置为1K,即将文件大小小于或等于1K的文件限定为小文件,门限值也可以设置为2M、4M等,本发明并不限于此,则文件系统可以从接收到的文件调用请求消息中确定小文件的调用请求消息,这些小文件的调用请求消息用于向文件系统请求调用对应的小文件。

[0079] 在本发明实施例中,文件系统确定L个第一文件调用请求消息,可以一次接受一个第一文件调用请求消息,也可以一次接收多个第一文件调用请求消息,例如,文件系统可以一次接收3个第一文件调用请求消息,则文件系统确定L个第一文件调用请求消息,即确定接收到的3个第一文件调用请求消息。也可以接收随机的第一文件调用请求消息,当积累到L个时,即确定L个第一文件调用请求消息,该L的值可以根据实际应用确定,也可以根据经验值设置,本发明并不限于此。

[0080] 可选地,在本发明实施例中,存取文件的装置200的确定模块210确定的该L个第一

文件调用请求消息中每个第一文件调用请求消息中可以包括物理扇区信息,该物理扇区信息包括存储该每个第一文件的调用请求消息请求调用的第一文件的物理扇区的地址。可选地,该物理扇区信息中具体可以包括请求调用的第一文件的起始地址、被调用的第一文件的大小以及下发该第一文件调用请求消息的设备的相关信息等。文件系统可以根据包括该物理扇区信息的第一文件调用请求消息确定要调用的第一文件。

[0081] 可选地,在本发明实施例中,确定模块210确定的该第一文件调用请求消息中还可以包括存取路径信息,该存取路径信息包括对应的存储该第一文件的物理扇区的路径信息,根据该存取路径信息可以在文件系统中找到存储对应的小文件的物理扇区。具体地,当打开open或关闭close一个文件或创建一个新文件时,文件系统的内核会分配一个文件描述符并返回给用户程序,该文件描述符表项中的指针指向新打开的文件。在存取文件时,用户程序把文件描述符传给内核,内核根据文件描述符找到相应的表项,再通过表项中的指针找到相应的文件。每个文件结构体都有一个指向dentry (directory entry,目录项) 结构体的指针,我们可以根据路径找到文件的inode节点。为了减少读盘次数,内核一般会缓存了目录的树状结构,称为dentry cache,其中每个inode节点都是是一个dentry结构体,只要沿着路径各部分的dentry搜索即可,dentry cache只保存最近访问过的目录项,如果没有命中,就要从磁盘读到内存中。

[0082] 具体地,对于小文件,文件系统可以将小文件的信息节点MF-inode以链表的形式进行存储,这些小文件的信息节点分别存储了对应的小文件的存取路径信息,可以根据信息节点中的存取路径信息查找小文件。当文件系统接收到第i个第一文件调用请求消息时,在存储的MF-inode链表中查找对应的第i个第一文件的信息节点,若该链表中存在该第i个第一文件的信息节点,则根据该第i个第一文件的信息节点,确定用于指示该第一文件的存储路径的存取路径信息,可以将该存取路径信息存储到该第i个第一文件调用请求消息中。当在MF-inode链表中无法查找到对应的第i个第一文件的信息节点,则申请该第i个第一文件对应的新的inode,并将该新的inode加入到MF-inode链表中,同时将该第一文件对应的存储路径和文件名保存到该新的inode中,记录为该inode的成员变量,也将该存取路径信息存储到该第i个第一文件调用请求消息中。

[0083] 可选地,对于小文件对应的MF-inode链表,当其中包括的inode对应的文件由于文件数据的不断写入,从小文件变为大文件时,例如当该小文件的大小超过4M时,可以通过调用截断(O_TRUNC)操作,使得大文件被截断为小文件;也可以将变大的文件移除小文件的MF-inode链表中,为其分配普通inode,由于性能上没有什么影响,也可以保持现状。相反地,当大文件变为小文件时,可以将该小文件添加到MF-inode链表,可选地,可以通过磁盘碎片整理程序(defragment daemon)在系统空闲的时候完成。

[0084] 在本发明实施例中,Linux的I/O调度程序是块设备I/O子系统的主要组件,它介于通用块层和块设备驱动程序之间,当Linux内核组件要读写一些数据时,并不是请求一发出,内核便立即执行该请求,而是将其推迟执行。延迟分配是块设备性能的关键机制,当传输一个新数据块时,内核检查能否通过扩展前一个一直处于等待状态的请求而满足新请求。

[0085] 具体地,每个块设备程序都有一个请求队列与之关联。在块设备初始化时,会分配并初始化请求队列。在这个时候,我们便可以为块设备驱动程序指定特定的I/O调度算法,

默认情况下是使用系统默认的调度算法。默认的调度算法会优先处理与上一个处理请求最近的请求,因而就会对某个请求忽略很长一段时间,即为电梯算法。为了避免这种情况,当请求被传给电梯算法是,就会启动超时定时器,从而来避免某个请求永远得不到处理的情况。为了提升用户体验,读请求将一般优先于写请求,因为读请求通常阻塞发出请求的进程。

[0086] 在本发明实施例中,可以采用这种算法对小文件的调用请求消息进行处理,可选地,还可以通过下面的I/O调度(scheduler)算法方法进行调度。在本发明实施例中,可以通过分配模块220根据L个第一文件调用请求消息中的物理扇区信息,将该L个第一文件调用请求消息分别添加到M个第一文件调度队列(MF dispatch Queue)中。文件系统为小文件的调用请求消息分配了对应的第一文件调度队列,对于文件系统接收到的第一文件调用请求消息,可以根据每个调用请求消息中的物理扇区信息,将L个第一文件调用请求消息分别添加到M个第一文件调度队列中,使得具有物理扇区地址连续的至少两个第一文件调用请求消息被添加到同一个第一文件调度队列中,其中,每个第一文件调度队列包括K个第一文件调用请求消息,K和M均为正整数。

[0087] 例如,在本发明实施例中,文件系统确定了6个第一文件调用请求消息,即 $L=6$,当只有一个第一文件调度队列,即小文件的调度队列时, $M=1$,而6个第一文件调用请求消息中存在三个第一文件调用请求消息中的物理扇区信息显示其物理扇区地址连续,另外三个不连续。将6个第一文件调用请求消息均放入该第一文件调度队列中,并将物理扇区地址连续的三个第一文件调用请求消息进行合并,即此时 $K=3$;而另外三个物理扇区地址不连续的第一文件调用请求消息暂时不做处理,待有新的第一文件调用请求消息时,再进行物理扇区地址是否连续的判断,若连续,则进行合并,如果不连续,则仍然不做处理。可选地,可以通过物理扇区地址是否连续来判断是否进行合并处理,还可以根据调用的第一文件的大小进行第一文件调用请求信息的合并,例如,当两个第一文件调用请求消息请求写入的两个第一文件的大小之和,恰好可以构成一个数据块(block),则也可以将这两个第一文件调用请求消息合并,并将对应的要求写入的第一文件写入同一个物理扇区内,同一个数据块中,本发明并不限于此。

[0088] 可选地,还例如,文件系统确定了6个第一文件调用请求消息,即 $L=6$,且将第一文件调度队列分为5个,即小文件的调度队列有5个, $M=5$ 时,而6个第一文件调用请求消息中存在三个第一文件调用请求消息中的物理扇区信息显示其物理扇区地址连续,另外三个不连续。则将该三个物理扇区地址连续的第一文件调用请求消息放入同一个第一文件调度队列中,例如将该三个第一文件调用请求消息放入第一个第一文件调度队列中,另外三个互相之间不连续的第一文件调用请求消息则可以放入另外三个第一文件调度队列中,而还有一个第一文件调度队列暂时为空。此时,第一个第一文件调度队列中存在3个第一文件调用请求消息,即 $K=3$,则可以对第一个第一文件调度队列中的三个第一文件调用请求消息进行合并处理,合并为一个第一文件调度队列的调用请求消息。而另外各有一个第一文件调用请求消息的三个第一调度队列则可以先不进行合并处理,继续等待分配其它第一文件调度请求消息,当这几个第一文件调度队列中存在多个可以合并的第一文件调用请求消息时,则同样地,与第一个第一文件调度队列一样,将在同一个第一文件调度队列中的多个第一文件调用请求消息进行合并。可选地,可以通过物理扇区地址是否连续来判断是否进行合并

处理,还可以根据调用的第一文件的大小进行第一文件调用请求信息的合并,例如,当两个第一文件调用请求消息请求写入的两个第一文件的大小之和,恰好可以构成一个数据块(block),则也可以将这两个第一文件调用请求消息合并,并将对应的要求写入的第一文件写入同一个物理扇区内,同一个数据块中,本发明并不限于此。

[0089] 可选地,又例如,文件系统确定了6个第一文件调用请求消息,即 $L=6$,如果存在3个第一文件调度队列,即 $M=3$,并且6个第一文件调用请求消息中三个第一文件调用请求消息中的物理扇区信息显示其物理扇区地址连续,另外三个不连续,则先将该三个物理扇区地址连续的第一文件调用请求消息放入同一个第一文件调度队列中,例如将该三个第一文件调用请求消息放入第一个第一文件调度队列中,而对于另外三个物理地址不连续的第一文件调用请求消息,可以将他们按照大小分配或随机分配到三个第一文件调度队列中;或者也可以根据他们请求调用的第一文件的存储地址的相关性,将他们分配到三个第一文件调度队列中;还可以将他们随机分配到三个第一文件调度队列中。然后对第一个第一文件调度队列中物理扇区信息连续的三个第一文件调用请求消息进行合并处理,合并为一个第一文件调度队列的调用请求消息。而另外的两个第一文件调度队列可以先不进行处理,或者第一个第一文件调度队列中如果还有没有被合并的第一文件调用请求消息,也暂时不进行处理,当这几个第一文件调度队列中存在多个可以合并的第一文件调用请求消息时,则同样地,与第一个第一文件调度队列一样,将在同一个第一文件调度队列中的多个第一文件调用请求消息进行合并。可选地,可以通过物理扇区地址是否连续来判断是否进行合并处理,还可以根据调用的第一文件的大小进行第一文件调用请求信息的合并,例如,当两个第一文件调用请求消息请求写入的两个第一文件的大小之和,恰好可以构成一个数据块(block),则也可以将这两个第一文件调用请求消息合并,并将对应的要求写入的第一文件写入同一个物理扇区内,同一个数据块中,本发明并不限于此。

[0090] 可选地,分配模块220可以按照 L 个第一文件调用请求消息的进程号(PID)的不同,将文件系统接收的调用请求消息进行分组,再将分组后的调用请求消息中的第一文件调用请求消息存储到第一文件的调度队列,该第一文件调度队列可以包括 M 个调度队列。可选地,可以在文件系统中新增一个结构体,用于判断哪些是第一文件调用请求消息;也可以在S110中确定第一文件调用请求消息时,将第一文件调用请求消息进行标识,通过该标识区分第一文件调用请求消息,并将第一文件调用请求消息分配到对应的 M 个第一文件调度队列中,每个第一文件调度对应可以包括 K 个第一文件调用请求消息。

[0091] 在本发明实施例中,文件系统在处理模块230针对 M 个第一文件调度队列中的每个第一文件调度队列,可以对其包括的 K 个第一文件调用请求消息进行处理,获得该每个第一文件调度队列的调用请求消息。具体地,将 L 个第一文件调用请求消息添加至 M 个第一文件调度队列,对于每个第一文件调度队列,当其中包括的至少两个第一文件调用请求消息,具有连续的物理扇区(bi_sector)地址,可以将该至少两个第一文件调用请求消息进行合并处理,使得该至少两个第一文件调用请求消息可以合成为一个调用请求消息,即第一文件调度队列的调用请求消息,从而减少了调用的次数。例如,对于任意一个第一文件调度队列,该第一文件调度队列包括 K 个第一文件调用请求消息,根据该 K 个第一文件调用请求消息中包括的物理扇区信息,可以对该 K 个第一文件调用请求消息进行处理,生成一个该第一文件调度队列的调用请求消息,该第一文件调度队列的调用请求消息用于调用 K 个第一文

件调用请求消息对应的第一文件。可选地,文件系统还可以对第一文件调用请求消息进行删除处理,删除不必要的调用请求,也可以减少调用次数。

[0092] 可选地,当某个第一文件调度队列只包括一个第一文件调用请求消息时,即不存在与该调用请求消息中物理扇区地址相连续的另一个调用请求消息时,可以暂时对该第一文件调用请求消息不作处理,当该第一文件的调度队列中包括至少两个第一文件调用请求消息的物理扇区地址连续时,再进行合并处理,本发明并不限于此。

[0093] 在本发明实施例中,文件系统的处理模块230将第一文件调度队列中包括的K个第一文件调用请求消息进行合并或删除处理后,调用模块240可以根据获得的第一文件调度队列的调用请求消息,存取相应的第一文件。具体地,由于每个第一文件调用请求消息中可以包括存取路径信息,在第一文件调度队列进行合并后,每个第一文件调度队列的调用请求消息仍然具有对应的存取路径信息,因此可以根据该存取路径信息,对应的对第一文件进行存取处理。可选地,第一文件调度队列的调用请求消息可以发送至存储介质请求队列(mmc_request)中,mmc_request处理第一文件调度队列的调用请求消息,并设置设备控制器,主要用于产生数据并传送,当数据传输完成后可以从对应的调度队列中删除当前的调用请求消息,并开始下一个请求对应的数据传输。

[0094] 因此,本发明实施例的存取文件的装置,通过确定小文件的调用请求消息,并为这些小文件的调用请求消息分配对应的第一文件调度队列,从而可以将多个小文件的调用请求消息进行合并和/或删除处理,减少小文件的调用次数,可以避免小文件的调用请求而增加的I/O次数并导致的占用过多内存资源,可以减少系统调用次数,并增加I/O请求的吞吐量,进而整体提升文件系统的I/O效率。

[0095] 可选地,作为一个实施例,根据本发明实施例的存取文件的装置200还包括:

[0096] 第二确定模块250,用于在存储文件的物理扇区上,确定N个第一文件存储空间组,该N个第一文件存储空间组用于存储第一文件。

[0097] 在本发明实施例中,具体地,文件系统可以为小文件的存取设置单独的物理扇区,该物理扇区中包括多个单独的区域,该区域包括多个第一文件存储空间组(MF Group),该区域的大小可以根据时隙应用进行调整及优化,该区域内包括的MF Groups的个数可以根据需求动态调整。可选地,对于单独存放小文件的多个MF Group,每个MF Group可以包括多个数据库(data block),每个data block可以存储一个或多个小文件,本发明并不限于此。

[0098] 具体地,第二确定模块250可以通过磁盘碎片整理程序(defragment daemon)在系统空闲时完成多个MF Group的合并和/或删除。例如,可以将多个MF Group进行合并,保证多个第一文件存储在连续的物理扇区上,进而可以更好地利用eMMC的特性。在系统的运行的过程中,由于文件的删除操作,使得某些MF Group中用于存储文件的空间有剩余,即该MF Group中大部分空间被加入了待删除的列表中,可以将这些MF Group进行合并,将至少两个MF Group中的数据合并存储到一个或多个MF Group中,从而释放出更多的空间,提升空间的利用率。

[0099] 可选地,作为一个实施例,也可以对MF Groups进行删除处理。例如,当MF Group中存在某些小文件变成大文件时,例如超过了4M的范围,可以找到该变大的文件所在的MF Group进行标记,然后当defragment daemon遍历待删除的列表,发现该MF Group中包括的第一文件均被加入到删除列表,使得该MF Group中已经不存在文件时,即该MF Groups为空

时,会进行删除,删除该MF Group来释放资源。

[0100] 可选地,作为一个实施例,也可以对MF Group进行增加处理。例如,当新增加了小文件时,例如:某个大文件变成小文件时,defragment daemon首先会在多个MF Group中查找存在剩余空间的MF Group,如果存在可用存储空间可以将该小文件存到该MF Group中,则将新增加的小文件存储到该MF Group中;如果不存在有剩余空间的MF Groups,defragment daemon会增加新的MF Group,用来存放该小文件以及其它新增加的小文件。

[0101] 在本发明实施例,文件系统的调用模块240可以根据该至少一个第一文件调度队列的调用请求消息,在存储文件的MF Group中查找需要存取调用的第一文件。

[0102] 应理解,根据本发明实施例的存取文件的装置200可对应于执行本发明实施例中的方法100,并且存取文件的装置200中的各个模块的上述和其它操作和/或功能分别为了实现图1和图2中的各个方法的相应流程,为了简洁,在此不再赘述。

[0103] 因此,本发明实施例的存取文件的装置,通过确定小文件的调用请求消息,并为这些小文件的调用请求消息分配对应的第一文件调度队列,从而可以将多个小文件的调用请求消息进行合并和/或删除处理,可以避免小文件的调用请求而增加的I/O次数并导致的占用过多内存资源,可以减少系统调用次数,并增加I/O请求的吞吐量,进而整体提升文件系统的I/O效率。另外,还可以为小文件分配相应的动态存储区域,将不同的小文件聚合在一起存放,可以减少存储空间的浪费,提升空间的利用率。

[0104] 如图5所示,本发明实施例还提供了一种存储系统300,包括处理器310、存储器320和总线系统330。其中,处理器310和存储器320通过总线系统330相连,该存储器320用于存储指令,该处理器310用于执行该存储器320存储的指令。该存储器320存储程序代码,且处理器310可以调用存储器320中存储的程序代码执行以下操作:确定L个第一文件调用请求消息,该L个第一文件调用请求消息中的每个第一文件调用请求消息用于请求调用第一文件,该第一文件的大小小于或等于门限值,该每个第一文件调用请求消息包括物理扇区信息,该物理扇区信息包括存储该第一文件的物理扇区的地址,L为大于1的整数;根据该物理扇区信息,将该L个第一文件调用请求消息分别添加到与该L个第一文件调用请求消息对应的M个第一文件调度队列中,M为正整数;对该M个第一文件调度队列中的每个第一文件调度队列包括的K个第一文件调用请求消息进行处理,获得该每个第一文件调度队列的调用请求消息,该每个第一文件调度队列的调用请求消息用于请求调用该K个第一文件调用请求消息对应的第一文件,K为正整数;根据该每个第一文件调度队列的调用请求消息,存取该K个第一文件调用请求消息对应的第一文件。

[0105] 因此,本发明实施例的存储系统,通过确定小文件的调用请求消息,并为这些小文件的调用请求消息分配对应的第一文件调度队列,从而可以将多个小文件的调用请求消息进行合并和/或删除处理,可以避免小文件的调用请求而增加的I/O次数并导致的占用过多内存资源,可以减少系统调用次数,并增加I/O请求的吞吐量,进而整体提升文件系统的I/O效率。另外,还可以为小文件分配相应的动态存储区域,将不同的小文件聚合在一起存放,可以减少存储空间的浪费,提升空间的利用率。

[0106] 应理解,在本发明实施例中,该处理器310可以是中央处理单元(Central Processing Unit,简称为“CPU”),该处理器310还可以是其他通用处理器、数字信号处理器(DSP)、专用集成电路(ASIC)、现成可编程门阵列(FPGA)或者其他可编程逻辑器件、分立门

或者晶体管逻辑器件、分立硬件组件等。通用处理器可以是微处理器或者该处理器也可以是任何常规的处理器等。

[0107] 该存储器320可以包括只读存储器和随机存取存储器,并向处理器310提供指令和数据。存储器320的一部分还可以包括非易失性随机存取存储器。例如,存储器320还可以存储设备类型的信息。

[0108] 该总线系统330除包括数据总线之外,还可以包括电源总线、控制总线和状态信号总线等。但是为了清楚说明起见,在图中将各种总线都标为总线系统330。

[0109] 在实现过程中,上述方法的各步骤可以通过处理器310中的硬件的集成逻辑电路或者软件形式的指令完成。结合本发明实施例所公开的方法的步骤可以直接体现为硬件处理器执行完成,或者用处理器中的硬件及软件模块组合执行完成。软件模块可以位于随机存储器,闪存、只读存储器,可编程只读存储器或者电可擦写可编程存储器、寄存器等本领域成熟的存储介质中。该存储介质位于存储器320,处理器310读取存储器320中的信息,结合其硬件完成上述方法的步骤。为避免重复,这里不再详细描述。

[0110] 可选地,作为一个实施例,处理器310可以调用存储器320中存储的程序代码执行以下操作:在存储文件的物理扇区上,确定N个第一文件存储空间组,该N个第一文件存储空间组用于存储第一文件,N为正整数;当该N个第一文件存储空间组中至少两个第一文件存储空间组存在可用存储空间时,将该至少两个第一文件存储空间组包括的第一文件进行合并;当该N个第一文件存储空间组中至少一个第一文件存储空间组不存储文件时,释放该至少一个第一文件存储空间组;当该N个第一文件的存储空间组都不存在该可用存储空间时,增加该N个第一文件的存储空间组的个数;根据该每个第一文件调度队列的调用请求消息,从该N个第一文件存储空间组中存取该K个第一文件调用请求消息对应的第一文件。

[0111] 可选地,作为一个实施例,处理器310可以调用存储器320中存储的程序代码执行以下操作:根据该物理扇区信息,将该L个第一文件调用请求消息中的至少两个第一文件调用请求消息添加到同一个第一文件调度队列中,该至少两个第一文件调用请求消息请求调用的第一文件的物理扇区的地址连续。

[0112] 可选地,作为一个实施例,该每个第一文件调用请求消息还包括存取路径信息,该存取路径信息用于指示存取该每个第一文件调用消息对应的第一文件的物理扇区的路径。

[0113] 可选地,作为一个实施例,处理器310可以调用存储器320中存储的程序代码执行以下操作:确定L个第一文件信息节点,该L个第一文件信息节点中的每个第一文件信息节点用于存储该每个第一文件调用请求消息请求调用的第一文件的该存取路径信息;根据该每个第一文件信息节点,确定该每个第一文件调用请求消息请求调用的第一文件的该存取路径信息。

[0114] 可选地,作为一个实施例,处理器310可以调用存储器320中存储的程序代码执行以下操作:在信息节点链表中确定是否存在该L个第一文件信息节点中第i个第一文件信息节点;当该信息节点链表中不存在该第i个第一文件信息节点时,在该信息节点链表中创建该第i个第一文件信息节点,并将第i个第一文件的该存取路径信息存储到该第i个第一文件信息节点中;当该信息节点链表中存在该第i个第一文件信息节点时,根据该第i个第一文件信息节点确定该第i个第一文件的该存取路径信息。

[0115] 因此,本发明实施例的存储系统,通过确定小文件的调用请求消息,并为这些小文

件的调用请求消息分配对应的第一文件调度队列,从而可以将多个小文件的调用请求消息进行合并和/或删除处理,可以避免小文件的调用请求而增加的I/O次数并导致的占用过多内存资源,可以减少系统调用次数,并增加I/O请求的吞吐量,进而整体提升文件系统的I/O效率。另外,还可以为小文件分配相应的动态存储区域,将不同的小文件聚合在一起存放,可以减少存储空间的浪费,提升空间的利用率。

[0116] 本领域普通技术人员可以意识到,结合本文中所公开的实施例描述的各示例的单元及算法步骤,能够以电子硬件、或者计算机软件和电子硬件的结合来实现。这些功能究竟以硬件还是软件方式来执行,取决于技术方案的特定应用和设计约束条件。专业技术人员可以对每个特定的应用来使用不同方法来实现所描述的功能,但是这种实现不应认为超出本发明的范围。

[0117] 所属领域的技术人员可以清楚地了解到,为描述的方便和简洁,上述描述的系统、装置和单元的具体工作过程,可以参考前述方法实施例中的对应过程,在此不再赘述。

[0118] 在本申请所提供的几个实施例中,应该理解到,所揭露的系统、装置和方法,可以通过其它的方式实现。例如,以上所描述的装置实施例仅仅是示意性的,例如,所述单元的划分,仅仅为一种逻辑功能划分,实际实现时可以有另外的划分方式,例如多个单元或组件可以结合或者可以集成到另一个系统,或一些特征可以忽略,或不执行。另一点,所显示或讨论的相互之间的耦合或直接耦合或通信连接可以是通过一些接口,装置或单元的间接耦合或通信连接,可以是电性,机械或其它的形式。

[0119] 所述作为分离部件说明的单元可以是或者也可以不是物理上分开的,作为单元显示的部件可以是或者也可以不是物理单元,即可以位于一个地方,或者也可以分布到多个网络单元上。可以根据实际的需要选择其中的部分或者全部单元来实现本实施例方案的目的。

[0120] 另外,在本发明各个实施例中的各功能单元可以集成在一个处理单元中,也可以是各个单元单独物理存在,也可以两个或两个以上单元集成在一个单元中。

[0121] 所述功能如果以软件功能单元的形式实现并作为独立的产品销售或使用,可以存储在一个计算机可读取存储介质中。基于这样的理解,本发明的技术方案本质上或者说对现有技术做出贡献的部分或者该技术方案的部分可以以软件产品的形式体现出来,该计算机软件产品存储在一个存储介质中,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)执行本发明各个实施例所述方法的全部或部分步骤。而前述的存储介质包括:U盘、移动硬盘、只读存储器(ROM,Read-Only Memory)、随机存取存储器(RAM,Random Access Memory)、磁碟或者光盘等各种可以存储程序代码的介质。

[0122] 以上所述,仅为本发明的具体实施方式,但本发明的保护范围并不局限于此,任何熟悉本技术领域的技术人员在本发明揭露的技术范围内,可轻易想到变化或替换,都应涵盖在本发明的保护范围之内。因此,本发明的保护范围应所述以权利要求的保护范围为准。

100

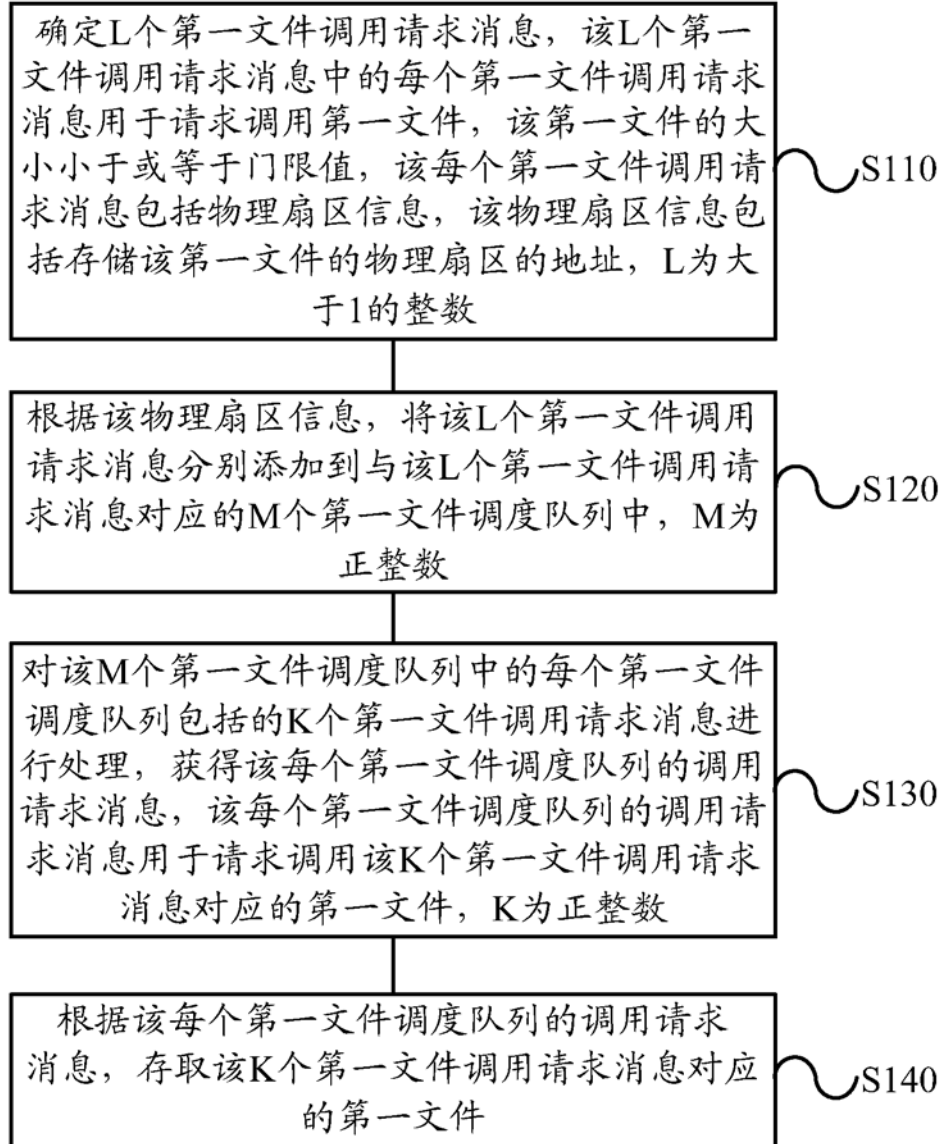


图1

100

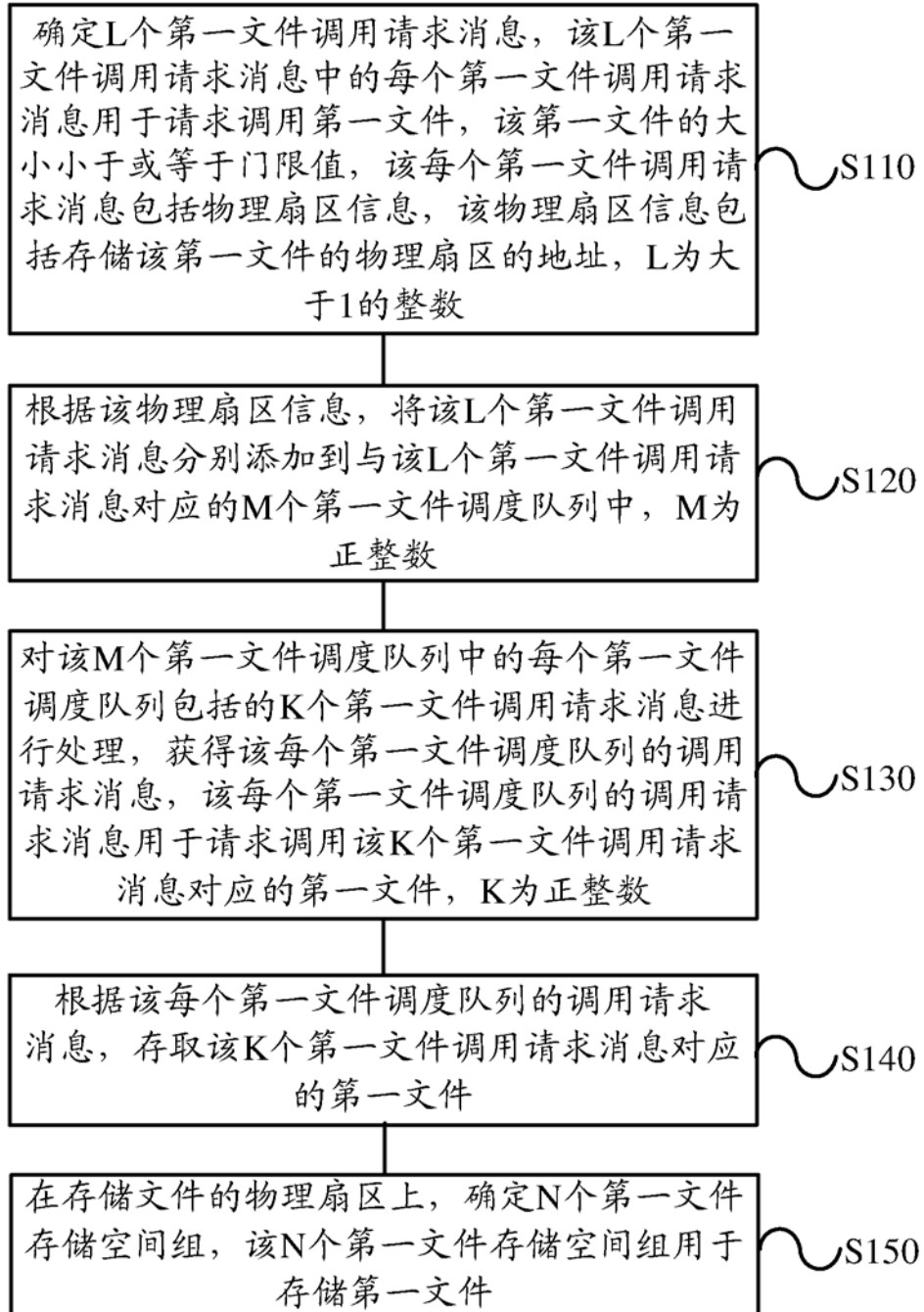


图2

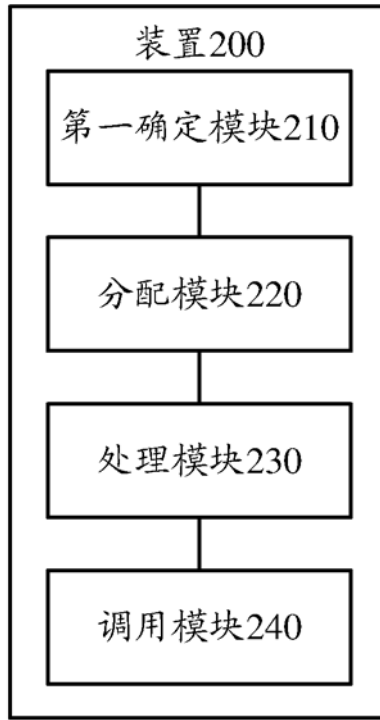


图3

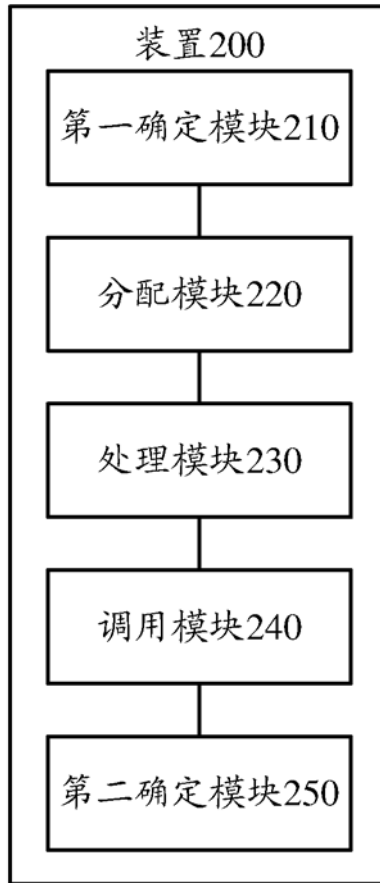


图4

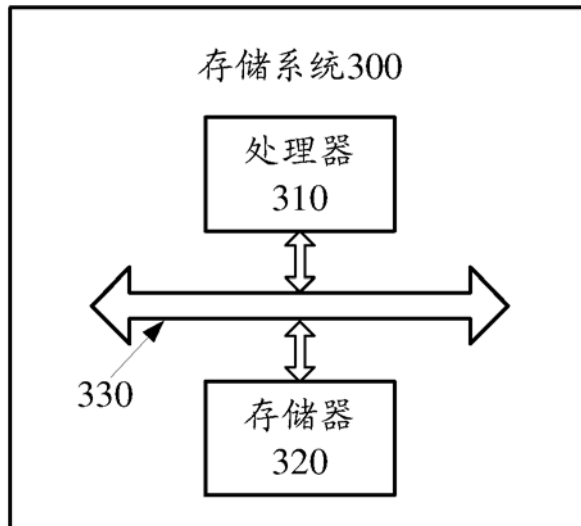


图5