

(12) 发明专利

(10) 授权公告号 CN 102156753 B

(45) 授权公告日 2012. 11. 14

(21) 申请号 201110110264. 9

US 2009144488 A1, 2009. 06. 04, 全文 .

(22) 申请日 2011. 04. 29

CN 102012873 A, 2011. 04. 13,

CN 101692229 A, 2010. 04. 07,

(73) 专利权人 中国人民解放军国防科学技术大学

审查员 权聪敏

地址 410073 湖南省长沙市砚瓦池正街 47 号中国人民解放军国防科学技术大学计算机学院软件研究所

(72) 发明人 戴华东 吴庆波 颜跃进 朱浩 孔金珠 秦莹

(74) 专利代理机构 湖南兆弘专利事务所 43008 代理人 周长清 谭武艺

(51) Int. Cl.

G06F 17/30 (2006. 01)

(56) 对比文件

CN 1728114 A, 2006. 02. 01, 全文 .

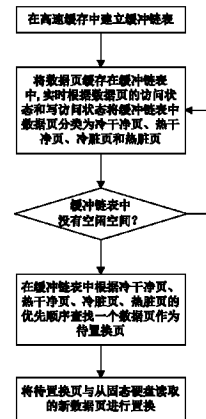
权利要求书 2 页 说明书 8 页 附图 4 页

(54) 发明名称

面向固态硬盘文件系统的数据页缓存方法

(57) 摘要

本发明公开了一种面向固态硬盘文件系统的数据页缓存方法,其实施步骤如下:1) 在高速缓存中建立用于缓存数据页的缓冲链表;2) 将固态硬盘中读取的数据页缓存在缓冲链表中以供访问,并实时根据数据页的访问状态和写访问状态将缓冲链表中数据页分类为冷干净页、热干净页、冷脏页和热脏页;3) 当缓冲链表中没有空闲空间时,首先在缓冲链表中根据冷干净页、热干净页、冷脏页、热脏页的优先顺序查找一个数据页作为待置换页,然后将待置换页与从固态硬盘读取的新数据页进行置换。本发明可充分利用固态硬盘特性、有效缓解外部存储的性能瓶颈、提高系统存储处理性能,具有 I/O 性能好、缓存页置换代价低、开销小、命中率高的优点。



1. 一种面向固态硬盘文件系统的数据页缓存方法,其特征在于其实施步骤如下:

1) 在高速缓存中建立用于缓存数据页的缓冲链表;

2) 将固态硬盘中读取的数据页缓存在缓冲链表中以供访问,并实时根据数据页的访问状态和写访问状态将缓冲链表中数据页分类为冷干净页、热干净页、冷脏页和热脏页;

3) 当缓冲链表中没有空闲空间时,首先在缓冲链表中根据冷干净页、热干净页、冷脏页、热脏页的优先顺序查找一个数据页作为待置换页,然后将待置换页与从固态硬盘读取的新数据页进行置换;

所述步骤 1) 中建立的缓冲链表包括用于缓存冷干净页的冷干净链表和用于缓存热干净页、冷脏页、热脏页的混合链表;

所述步骤 2) 中首先为缓冲链表中的数据页建立用于标记访问状态的冷热标志、设置冷热标志的初始值、冷阈值范围和热阈值范围,当数据页首次缓存至缓冲链表中时将数据页的冷热标志置为初始值,缓冲链表中的数据页每发生一次访问则冷热标志向热阈值范围变化、每经过一次待置换页查找则冷热标志向冷阈值范围变化,所述进行数据页分类时,将冷热标志属于冷阈值范围内且未发生写访问的数据页分类为冷干净页、将冷热标志属于冷阈值范围内且已发生写访问的数据页分类为冷脏页、将冷热标志属于热阈值范围内且未发生写访问的数据页分类为热干净页、将冷热标志属于热阈值范围内且已发生写访问的数据页分类为热脏页;所述进行数据页分类时,如果数据页为冷干净页或者冷脏页且被再次访问,则修改该数据页的冷热标志使其属于热阈值范围内;所述冷干净链表中的数据页一旦发生访问操作,则将其转移至混合链表;所述冷干净链表和混合链表中的数据页采用最近最少使用算法进行实时排序,将缓存至冷干净链表的数据页写入冷干净链表的表头,冷干净链表中的数据页转移至混合链表时写入混合链表的表头,在冷干净链表中查找待置换页时,从冷干净链表的表尾向表头方向查找,在混合链表中查找待置换页时,从混合链表的表尾向表头方向查找待置换页;

所述步骤 3) 中,首先在冷干净链表中查找待置换页,如果冷干净链表为空,则在混合链表中根据热干净页、冷脏页、热脏页的优先顺序查找待置换页。

2. 根据权利要求 1 所述的面向固态硬盘文件系统的数据页缓存方法,其特征在于:所述从混合链表的表尾向表头方向查找待置换页时,首先在混合链表的表尾建立一个用于查找热干净页的查找窗口,然后首先在查找窗口内查找热干净页,如果找到热干净页则将该热干净页作为待置换页,如果在查找窗口内没有找到热干净页,则首先从混合链表的表尾向表头方向查找冷脏页作为待置换页,如果在混合链表中没有找到冷脏页,则将混合链表的表尾端的热脏页作为待置换页。

3. 根据权利要求 2 所述的面向固态硬盘文件系统的数据页缓存方法,其特征在于:所述建立查找窗口后,实时根据数据缓存的当前主要访问类型调整查找窗口的窗口大小,如果当前主要访问类型为读访问,则减少混合链表中查找窗口的窗口大小;如果当前主要访问类型为写访问,则增加混合链表中查找窗口的窗口大小。

4. 根据权利要求 3 所述的面向固态硬盘文件系统的数据页缓存方法,其特征在于:所述步骤 1) 中建立用于保存被置换数据页信息的克隆列表、设置用于判断缓冲链表主要访问类型的读密集阈值和写密集阈值;所述步骤 3) 中进行置换时将被置换数据页信息保存在克隆列表中,同时实时检查克隆列表中的所有数据页,如果克隆列表中冷干净页和热干净

页数量之和占克隆列表中数据页总数的比例超过读密集阈值,则判定缓冲链表当前主要访问类型为读访问;如果克隆列表中冷脏页和热脏页数量之和占克隆列表中数据页总数的比例超过写密集阈值,则判定缓冲链表当前主要访问类型为写访问。

5. 根据权利要求3或4所述的面向固态硬盘文件系统的数据页缓存方法,其特征在于:所述减少窗口大小后查找窗口的窗口大小 W 为 $W=(1-1/(\mu+1)) \cdot W_0$,所述增加窗口大小后查找窗口的窗口大小 W 为 $W=(1+1/(\mu+1)) \cdot W_0$,其中 W_0 为查找窗口的初始窗口大小, μ 为查找窗口的窗口大小 W 与缓冲链表总数据页数量的比值。

面向固态硬盘文件系统的缓存方法

技术领域

[0001] 本发明涉及文件系统缓存机制,特指面向固态硬盘的文件系统数据页的缓存管理方法。

背景技术

[0002] 当前处理器性能一直按照摩尔定律在飞速提高,相对而言,以机械磁盘为主的存储系统性能提高却缓慢得多,计算性能和存储性能之间的差距越来越大,存储系统的 I/O 性能瓶颈问题日益突出。磁盘作为当前主流的外部存储设备是提高整个计算机系统性能的一个瓶颈所在,因为它的读写访问速度和内存相比一般都要相差 3 个数量级,在磁盘上读取数据会影响整个应用的运行速度。目前,基于 Flash 的固态硬盘在 I/O 性能上有很大的上升空间,有很大的研究潜力,和磁盘相比固态硬盘有很多明显的优点:很小的读延迟,快速随机读写访问,低功耗,可靠性高等,所以固态硬盘比传统磁盘的应用前景更加广阔,并且随着固态存储制造工艺会越来越成熟,性能会越来越高的,这些都可能促使固态硬盘会逐步取代目前的磁盘。

[0003] 在存储系统中使用固态硬盘虽然能够提供比传统机械磁盘更高的读写访问性能,但是由于操作系统在软件设计和优化中假定的存储系统不是固态硬盘,使得操作系统没有完全发挥固态硬盘本身的优势,固态硬盘的性能没有达到最理想的状态。提高存储系统性能的一般方法是使用文件系统缓存,内存作为缓存位于文件系统和存储设备之间,能减少从文件系统到存储的读和写请求数目,提高存储 I/O 性能。缓存的性能直接影响文件系统的性能,必须高效地维护缓存。当缓存中没有空闲空间时,为了满足接下来的读写缓存请求必须按照某种机制置换出一些数据,这就牵涉到缓存的数据页置换机制。大部分的传统的数据页置换方法主要关注的缓存的命中率,因为命中率越高,减少读写次数越多,磁盘读写请求开销差不多,进行置换时内存中的读写数据不区别看待。

[0004] 文件系统的缓存都是通过一个用于缓存数据页的缓冲链表来实现的,目前面向固态硬盘文件系统的缓存方法为 CFLRU(干净页优先最近最少使用,Clean First LRU)法和 LRU-WSR(最近最少使用写操作冲排序,LRU-Write Sequence Recordering)法这两种方法:

[0005] 如图 1 所示,CFLRU 法将缓冲链表分为工作区和干净优先区,根据数据页的访问状态将数据页分为干净页和脏页。工作区缓存最近使用的数据页,干净页基本上都位于这个区域,所以大部分的缓冲命中在这个区域;干净优先区是进行页置换的区域,干净优先区的大小(包含缓存页的数目)用窗口大小 W 来衡量,窗口大小 W 是固定不变的。在干净优先区选择待置换页时,则在干净优先区的窗口大小 W 范围内从表尾向表头方向遍历直到选中了一个干净页,那么这个干净页就是待置换页,然后将待置换页进行置换写回。本图中 D 表示脏页, C 表示干净页, $P1$ 为表头端的数据页, $P8$ 为表尾端的数据页,从图中可以可见 $P7$ 为窗口大小 W 范围内从表尾向表头方向的第一个干净页,因此此时 $P7$ 为待置换页。

[0006] 如图 2 所示,LRU-WSR 法结合数据页的访问频率,对缓冲链表的数据页采用冷热标

志对脏页根据访问频率进行简单分类：干净页 C、热脏页 HD 和冷脏页 CD。P1 为表头端的数据页，P7 为表尾端的数据页，LRU-WSR 法的置换过程如下：先从表尾端选择一个数据页作为候选页，若候选页是干净页，那么不考虑冷热标志就选择它来置换。若候选页是脏页，必须先要检查它的冷热标志：如果它的冷热标志被设置，则选择这个脏页作为待置换页进行置换写回；如果冷热标志没有被设置，说明冷热标志为热，然后将数据页移动至表头，同时设置冷热标志。相当于一个脏页被给了二次机会，同时会从表尾端的选择另一个候选页进行再次判断，设置了脏页的冷热标志被在此引用时会被清除的，同时移动到表头，图中可知 P7 是干净页，因此此时 P7 会被选择作为待置换页进行置换。

[0007] 但是在 Flash 存储系统中，读和写开销完全不对称，写的访问时间一般都是读的 10 倍以上，而上述方法均没有考虑固态硬盘的读写不对称性，无法发挥固态硬盘的特性，不能有效发挥软件管理高速缓存的优势，缓存利用率不高，无法很好缓解存储的 I/O 性能瓶颈。

发明内容

[0008] 本发明要解决的技术问题是：提供一种可充分利用固态硬盘特性、有效缓解外部存储的性能瓶颈、提高系统存储处理性能，I/O 性能好、缓存页置换代价低、开销小、命中率高的面向固态硬盘文件系统的数据页缓存方法。

[0009] 为了解决上述技术问题，本发明采用的技术方案为：一种面向固态硬盘文件系统的数据页缓存方法，其实施步骤如下：

[0010] 1) 在高速缓存中建立用于缓存数据页的缓冲链表；

[0011] 2) 将固态硬盘中读取的数据页缓存在缓冲链表中以供访问，并实时根据数据页的访问状态和写访问状态将缓冲链表中数据页分类为冷干净页、热干净页、冷脏页和热脏页；

[0012] 3) 当缓冲链表中没有空闲空间时，首先在缓冲链表中根据冷干净页、热干净页、冷脏页、热脏页的优先顺序查找一个数据页作为待置换页，然后将待置换页与从固态硬盘读取的新数据页进行置换。

[0013] 作为本发明的进一步改进：

[0014] 所述步骤 2) 中首先为缓冲链表中的数据页建立用于标记访问状态的冷热标志、设置冷热标志的初始值、冷阈值范围和热阈值范围，当数据页首次缓存至缓冲链表中时将数据页的冷热标志置为所述初始值，缓冲链表中的数据页每发生一次访问则冷热标志向热阈值范围变化、每经过一次待置换页查找则冷热标志向冷阈值范围变化，步骤 2) 中进行数据页分类时，将冷热标志属于冷阈值范围内且未发生写访问的数据页分类为冷干净页、将冷热标志属于冷阈值范围内且已发生写访问的数据页分类为冷脏页、将冷热标志属于热阈值范围内且未发生写访问的数据页分类为热干净页、将冷热标志属于热阈值范围内且已发生写访问的数据页分类为热脏页。

[0015] 所述步骤 2) 中进行数据页分类时，如果数据页为冷干净页或者冷脏页且被再次访问，则修改该数据页的冷热标志使其属于热阈值范围内。

[0016] 所述步骤 1) 中建立的缓冲链表包括用于缓存冷干净页的冷干净链表和用于缓存热干净页、冷脏页、热脏页的混合链表；所述步骤 2) 中，冷干净链表中的数据页一旦发生访

问操作,则将其转移至混合链表;所述步骤3)中,首先在冷干净链表中查找待置换页,如果冷干净链表为空,则在混合链表中根据热干净页、冷脏页、热脏页的优先顺序查找待置换页。

[0017] 所述步骤2)中冷干净链表和混合链表中的数据页采用最近最少使用算法进行实时排序,将缓存至冷干净链表的数据页写入冷干净链表的表头,冷干净链表中的数据页转移至混合链表时写入混合链表的表头,在冷干净链表中查找待置换页时,从冷干净链表的表尾向表头方向查找,在混合链表中查找待置换页时,从混合链表的表尾向表头方向查找待置换页。

[0018] 所述从混合链表的表尾向表头方向查找待置换页时,首先在混合链表的表尾建立一个用于查找热干净页的查找窗口,然后首先在查找窗口内查找热干净页,如果找到热干净页则将该热干净页作为待置换页,如果在查找窗口内没有找到热干净页,则首先从混合链表的表尾向表头方向查找冷脏页作为待置换页,如果在混合链表中没有找到冷脏页,则将混合链表的表尾端的热脏页作为待置换页。

[0019] 所述建立查找窗口后,实时根据数据缓存的当前主要访问类型调整查找窗口的窗口大小,如果当前主要访问类型为读访问,则减少混合链表中查找窗口的窗口大小;如果当前主要访问类型为写访问,则增加混合链表中查找窗口的窗口大小。

[0020] 所述步骤1)中一并建立的用于保存被置换数据页信息的克隆列表、设置用于判断缓冲链表主要访问类型的读密集阈值和写密集阈值;所述步骤3)中进行置换时将被置换数据页信息保存在克隆列表中,同时实时检查克隆列表中的所有数据页,如果克隆列表中冷干净页和热干净页数量之和占克隆列表中数据页总数的比例超过读密集阈值,则判定缓冲链表当前主要访问类型为读访问;如果克隆列表中冷脏页和热脏页数量之和占克隆列表中数据页总数的比例超过写密集阈值,则判定缓冲链表当前主要访问类型为写访问。

[0021] 所述减少窗口大小后查找窗口的窗口大小 W 为 $W = (1 - 1/(\mu + 1)) \cdot W_0$, 所述增加窗口大小后查找窗口的窗口大小 W 为 $W = (1 + 1/(\mu + 1)) \cdot W_0$, 其中 W_0 为查找窗口的初始窗口大小, μ 为查找窗口的窗口大小 W 与缓冲链表总数据页数量的比值。

[0022] 本发明具有下述优点:

[0023] 1、本发明将缓冲链表中数据页分类为冷干净页、热干净页、冷脏页和热脏页,然后在进行数据页置换时根据冷干净页、热干净页、冷脏页、热脏页的优先顺序查找待置换页,优先置换开销小的冷干净页,尽量延迟置换开销大的热脏页,缓存整体开销小,具有可有效缓解外部存储的性能瓶颈、可提高系统存储处理性能、I/O 性能好、缓存页置换代价低、开销小、命中率高的优点。

[0024] 2、本发明通过冷热标志来对数据页进行分类、把数据页分为多个冷热等级的数据页冷检测机制,通过对数据页冷热分类实现根据数据页访问频率的细粒度调整,简单方便、置换开销小。

[0025] 3、本发明进一步在进行数据页分类时,如果数据页为冷干净页或者冷脏页且被再次访问则修改该数据页的冷热标志使其属于热阈值范围内,使其直接变成热干净页或者热脏页,可以提高缓存中数据页的命中率。

[0026] 4、本发明的缓冲链表进一步包括用于缓存冷干净页的冷干净链表和用于缓存热干净页、冷脏页、热脏页的混合链表,因此优先查找冷干净链表获取冷干净页,在冷干净链

表为空时才去查找混合链表,因此查找置换页的开销更小、查找速度更快,可以进一步提高缓冲链表的访问性能。

[0027] 5、本发明冷干净链表和混合链表中的数据页进一步采用最近最少使用算法进行实时排序,查找待置换页的时候从表尾向表头方向查找,可以有效保障开销最大的热脏页总是最后才被置换,有效提高了缓冲链表的访问性能。

[0028] 6、本发明进一步在查找热干净页的时候在查找窗口中进行查找,可以防止热干净页的过度置换,通过有效结合考虑数据页的新近性和访问频率,置换开销小、性能高。

[0029] 7、本发明进一步实时根据数据缓存的当前主要访问类型调整查找窗口的窗口大小,可以减少对固态硬盘的写回操作数量,从而可以充分发挥固态硬盘读取比写回速度快的特性,可以有效提高固态硬盘的存储系统的 I/O 效率和文件系统缓存性能。

[0030] 8、本发明进一步通过克隆列表来检测数据缓存的当前主要访问类型,克隆列表仅仅保存被置换数据页的元数据信息,实现简单高效、占用空间很小。

附图说明

[0031] 图 1 为 CFLRU 法的缓冲链表结构示意图;

[0032] 图 2 为 LRU-WSR 法的缓冲链表结构示意图;

[0033] 图 3 为本发明实施例缓存方法的流程示意图;

[0034] 图 4 为本发明实施例中混合链表的结构示意图;

[0035] 图 5 为本发明实施例中冷干净链表的结构示意图;

[0036] 图 6 为本发明实施例缓存方法查找代置换页的详细流程示意图;

[0037] 图 7 为本发明实施例中数据页类型转换示意图;

[0038] 图 8 为本发明实施例中克隆链表与冷干净链表、混合链表的大小比较示意图。

具体实施方式

[0039] 如图 3 所示,本发明实施例面向固态硬盘文件系统的数据页缓存方法的实施步骤如下:

[0040] 1) 在高速缓存中建立用于缓存数据页的缓冲链表;

[0041] 2) 将固态硬盘中读取的数据页缓存在缓冲链表中以供访问,并实时根据数据页的访问状态和写访问状态将缓冲链表中数据页分类为冷干净页、热干净页、冷脏页和热脏页;

[0042] 3) 当缓冲链表中没有空闲空间时,首先在缓冲链表中根据冷干净页、热干净页、冷脏页、热脏页的优先顺序查找一个数据页作为待置换页,然后将待置换页与从固态硬盘读取的新数据页进行置换。

[0043] 本实施例中,高速缓存是指内存。从读写开销的角度考虑,由于固态硬盘的读写开销具有非对称性的特点,同等情况下置换脏页引起的写回操作开销肯定是最大的,而置换干净页的开销最小。而从缓冲命中率的角度考虑,置换热页引起的影响肯定比置换冷页的影响大,引起的性能损失严重,这是由于热页包含了应用中访问频繁的页面数据,如果选择它来置换,在随后的应用请求很有可能因为再次访问这个页面数据而发生缺页,必须到外部存储中取数据引起系统性能的损失,如果选择冷页来置换,因为它的访问频率相对比较

低,将来再次引用的可能性小,可以以较小的代价将其置换还不引起大的性能损失。根据读写开销和页面访问频率的分析,可以知道置换热脏页的开销最大,置换冷干净页的开销最小。因此本实施例中,在缓冲链表中查找待置换页时,是根据冷干净页、热干净页、冷脏页、热脏页的优先顺序进行依次查找。

[0044] 步骤 2) 中首先为缓冲链表中的数据页建立用于标记访问状态的冷热标志、设置冷热标志初始值、冷阈值范围和热阈值范围,当数据页首次缓存至缓冲链表中时将数据页的冷热标志置为初始值,缓冲链表中的数据页每发生一次访问则冷热标志向热阈值范围变化、每经过一次待置换页查找则冷热标志向冷阈值范围变化,步骤 2) 中进行数据页分类时,将冷热标志属于冷阈值范围内且未发生写访问的数据页分类为冷干净页、将冷热标志属于冷阈值范围内且已发生写访问的数据页分类为冷脏页、将冷热标志属于热阈值范围内且未发生写访问的数据页分类为热干净页、将冷热标志属于热阈值范围内且已发生写访问的数据页分类为热脏页。本实施例中冷热标志用 COOL 表示,冷阈值范围为 2 ~ 3,热阈值范围为 0 ~ 1,冷热标志为 2 或 3 说明该页是冷数据页(包括冷干净页和冷脏页),冷热标志为 0 或 1 时表示数据页是热数据页(包括热脏页和热干净页),COOL = 0 表明数据页最热;COOL = 3 表明数据页最大程度即访问频率最小或最久没有访问了,冷热标志的初始值为 $COOL_{MAX} = 3$,缓冲链表中的数据页每发生一次访问则冷热标志减 1、每经过一次待置换页查找则冷热标志向冷阈值范围加 1。此外也可以将 $COOL_{MAX}$ 初始设定为 1,这样 COOL = 0 表示热,COOL = $COOL_{MAX}$ 表示冷,冷热标志退化为只有冷热两种程度了,这是最简单的情况。除了本实施例中采用 0 表示最热、 $COOL_{MAX}$ 表示最冷的方式以外,也可以将 0 或者其他值表示最冷、 $COOL_{MAX}$ 表示最热,或者采用其他表示最冷或者最热的值,或者采用其他冷阈值范围和热阈值范围,又或者将冷阈值范围和热阈值范围采用同一个边界值进行区分等等,其本质与本实施例等同。通过对数据页访问频率进行优化,可以提高缓冲链表的命中率和 I/O 性能。

[0045] 步骤 2) 中进行数据页分类时,如果数据页为冷干净页或者冷脏页且被再次访问,则修改该数据页的冷热标志使其属于热阈值范围内。本实施例中,如果冷干净页被再次访问,则将冷干净页的冷热标志直接置为 0 成为热干净页;如果冷脏页被再次访问,则将冷脏页的冷热标志直接置为 0 成为热脏页。

[0046] 步骤 1) 中建立的缓冲链表包括用于缓存冷干净页的冷干净链表和用于缓存热干净页、冷脏页、热脏页的混合链表;步骤 2) 中,冷干净链表中的数据页一旦发生访问操作,则将其转移至混合链表;步骤 3) 中,首先在冷干净链表中查找待置换页,如果冷干净链表为空,则在混合链表中根据热干净页、冷脏页、热脏页的优先顺序查找待置换页。本实施例中,缓冲链表包括混合链表 L1 和冷干净链表 L2,混合链表 L1 和冷干净链表 L2 包含的数据页数量 L 总和即为文件系统缓存的大小。混合链表 L1 缓存热干净页、冷脏页、热脏页,其中热干净页以 HC 表示、冷脏页以 CD 表示、热脏页以 HD 表示;冷干净链表 L2 仅仅缓存冷干净页,其中冷干净页以 CC 表示。

[0047] 如图 4 和图 5 所示,混合链表 L1 中缓存有热干净页 HC、冷脏页 CD、热脏页 HD,其中位于表头端的数据页 P1 为热脏页 HD,位于表尾端的数据页 P7 为热脏页 HD。冷干净链表 L2 仅仅缓存冷干净页 CC,数据页 P9 位于冷干净链表 L2 的表头端,数据页 P15 位于冷干净链表 L2 的表尾端。

[0048] 步骤 2) 中冷干净链表和混合链表中的数据页采用最近最少使用算法(LRU 算法)

进行实时排序,将缓存至冷干净链表的数据页写入冷干净链表的表头,冷干净链表中的数据页转移至混合链表时写入混合链表的表头,在冷干净链表中查找待置换页时,从冷干净链表的表尾向表头方向查找,在混合链表中查找待置换页时,从混合链表的表尾向表头方向查找待置换页。本实施例实质上是在传统的缓冲链表的基础上增加了一个新的仅缓存冷干净页的冷干净链表,冷干净链表中的冷干净页根据最近最少使用算法(LRU算法)的顺序排列,查找窗口位于混合链表L1的表尾。此外,最近最少使用算法的表头也可叫做最近最多使用端,表尾也叫最久最少使用端。

[0049] 根据前面对各种类型页的置换开销分析,我们知道置换热脏页的代价最大,在进行置换时要尽可能保证热脏页延迟置换,因此在混合链表的查找窗口中,根据热干净页、冷脏页、热脏页的优先顺序依次查找待置换页,热脏页是作为最后的选择。

[0050] 本实施例从混合链表的表尾向表头方向查找待置换页时,首先在混合链表的表尾建立一个用于查找热干净页的查找窗口,然后首先在查找窗口内查找热干净页,如果找到热干净页则将该热干净页作为待置换页,如果在查找窗口内没有找到热干净页,则首先从混合链表的表尾向表头方向查找冷脏页作为待置换页,如果在混合链表中没有找到冷脏页,则将混合链表的表尾端的热脏页作为待置换页。本实施例中查找待置换页的流程如下:

[0051] 1) 首先查找冷干净链表L2,如果冷干净链表L2不为空,则选定冷干净链表L2表尾端的数据页(冷干净页)作为待置换页,否则执行下一步。

[0052] 2) 如果冷干净链表L2为空,则查找混合链表L1:

[0053] 2.1) 在混合链表L1的查找窗口内从表尾向表头方向查找热干净页,数据页每经过一次查找则冷热标志加1,如果找到热干净页则将该数据页作为待置换页,如果没有找到热干净页则进行下一步;

[0054] 2.2) 从混合链表L1的表尾向表头方向遍历查找冷脏页,如果找到冷脏页则将该数据页作为待置换页,如果没有找到冷脏页则进行下一步;

[0055] 2.3) 将查找窗口表尾端的数据页(热脏页)作为待置换页。

[0056] 如图6所示,本实施例中查找待置换页并进行置换的详细流程如下:

[0057] (1)、如果冷干净链表L2非空,则标记冷干净链表L2表尾端的数据页为待置换页、跳转至步骤(7),否则执行下一步;

[0058] (2)、将混合链表L1表尾端的数据页为候选页,然后执行下一步;

[0059] (3)、如果候选页为热干净页,则将该数据页作为待置换页并跳转至步骤(7),否则执行下一步;

[0060] (4)、从混合链表L1的表尾向表头方向遍历每一个查找窗口内的数据页来查找热干净页,如果找到热干净页则将该数据页作为待置换页并跳转至步骤(7),如果当前的候选页不是热干净页,则首先将候选页的冷热标志加1,然后判断该数据页是否位于查找窗口内,如果已经遍历查找窗口的所有数据页仍未找到热干净页则执行下一步;

[0061] (5)、从混合链表L1的表尾向表头方向遍历每一个数据页来查找冷脏页,如果找到冷脏页则将该数据页作为待置换页并跳转至步骤(7),如果最终没有找到冷脏页则执行下一步;

[0062] (6)、将查找窗口表尾端的数据页(热脏页)作为待置换页,然后执行步骤(7),

[0063] (7)、将待置换页进行置换,置换完毕后退出现本次置换流程。

[0064] 如图 7 所示,对于首次缓存至缓冲链表的数据页,冷热标志默认设置为 $COOL_{MAX}$ 并插入到冷干净链表 L2 的表头。因为冷干净链表 L2 更新速度非常快,如果冷干净链表 L2 中的冷干净页被读取,说明该数据页从缓存至冷干净链表 L2 到再次发生访问并转移至混合链表 L1 的两次访问时间间隔非常短,因此本实施例中直接将冷干净链表 L2 中再次发生访问数据页的冷热标志置为 0,使其成为热干净页以提高缓存中数据页的命中率,并立即被转移到混合链表 L1 的表头,这也是混合链表 L1 中热干净页的唯一来源;如果冷干净链表 L2 中的冷干净页发生写访问,冷热标志保持不变,冷干净页变为冷脏页,立即被转移到混合链表 L1 的表头。混合链表 L1 中,如果冷脏页再次发生写访问,则冷热标志减 1,如果冷热标志落入热阈值范围内,则冷脏页变为热脏页。对于首次写访问并缓存至缓冲链表的数据页,则直接作为冷脏页放入混合链表 L1 的表头,这种方式会产生插入至混合链表 L1 中的冷脏页。对于混合链表 L1 中的数据页,不管它当前处于什么状态,只要它被访问了,那么它的冷热标志都要减 1。同时在混合链表 L1 中查找待置换页时,如果候选页不是干净页,则将冷热标志加 1,如果热脏页的冷热标志落入冷阈值范围内则变为冷脏页,从而可以有效保证热脏页总是被延迟置换。

[0065] 本实施例在建立查找窗口后,实时根据数据缓存的当前主要访问类型调整查找窗口的窗口大小,如果当前主要访问类型为读访问,则减少混合链表中查找窗口的窗口大小;如果当前主要访问类型为写访问,则增加混合链表中查找窗口的窗口大小。若当前主要访问类型为读访问,则读命中就显得相当重要,若还是大量置换干净页就会极大的增加缓冲的失效率,本实施例通过减少查找窗口的窗口大小尽量避免干净页被替换,可以有效提高读命中的命中率;若当前主要访问类型为写访问,则写命中就显得相当重要,本实施例通过增加查找窗口的窗口大小置换出更多的干净页,从而尽可能保持脏页不被置换,可以有效提高写命中的命中率,通过对读写访问模式的检测,可以充分发挥出固态硬盘写访问比读访问快的特性,提高系统的 I/O 性能。

[0066] 本实施例的混合链表 L1 中查找窗口的窗口大小 W 是动态变化的,窗口大小变化的基本依据是缓冲链表当前主要访问类型,当前主要访问类型可以根据混合链表 L1 和冷干净链表 L2 中的数据页分类进行判断。但是本实施例中对当前主要访问类型的判断是通过克隆链表 L3 实现的,其实现方法如下:

[0067] 步骤 1) 中建立用于保存被置换数据页信息的克隆列表,设置用于判断缓冲链表主要访问类型的读密集阈值和写密集阈值,步骤 3) 中进行置换时将被置换数据页信息保存在克隆列表中,同时实时检查克隆列表中的所有数据页,如果克隆列表中冷干净页和热干净页数量之和占克隆列表中数据页总数的比例超过读密集阈值,则判定缓冲链表当前主要访问类型为读访问;如果克隆列表中冷脏页和热脏页数量之和占克隆列表中数据页总数的比例超过写密集阈值,则判定缓冲链表当前主要访问类型为写访问。总得来说,发生置换时文件系统缓存的总空间大小等于混合链表 L1 加上冷干净链表 L2 的总页面数据大小,但是不同的应用访问模式会产生不同的链表组织情况,使得混合链表 L1 和冷干净链表 L2 包含的页面数发生变化。当前主要访问类型为写访问,即表现为写密集时,大量的写访问所占空间会大于缓冲链表,写密集让冷页面快速成为热页,触发置换时一方面会使冷干净链表 L2 中的冷干净页转化为热干净页或者冷脏页进入混合链表 L1,另一方面一些冷干净页

会被优先置换掉,使得冷干净链表 L2 完全为空,文件缓存的所有数据页全部都存储到混合链表 L1 中。但是,当访问是读访问时,冷干净链表 L2 发生置换的频率会比较高,为空的概率就会很小。

[0068] 如 8 所示,本实施例中混合链表 L1 的数据页数量 $|L1|$ 和冷干净链表 L2 的数据页数量 $|L2|$ 之和 $|L1|+|L2|$ 最大为 L,克隆链表 L3 的数据页数量 $|L3|$ 最大也为 L,其中 L 指文件系统缓冲的大小。混合链表 L1、冷干净链表 L2、克隆链表 L3 都是采用最近最少使用算法 (LRU 算法) 进行实时排序,即最近访问数据页放在表头 (LRU 算法的最近最多使用端),最久没有访问的依次往表尾 (LRU 算法的最久最少使用端)。除了优先选择冷干净页进行置换外,本实施例实现了查找窗口的窗口大小的动态调整,既可以说像 CFLRU 法里那样操作混合链表 L1,而且窗口大小还可以根据缓冲链表的当前主要访问类型而发生动态改变以取得最好的性能效果。

[0069] 为了实现对当前主要访问类型的判断,本实施例借助克隆链表实现,克隆链表 L3 中仅仅包含混合链表 L1 和冷干净链表 L2 中被置换数据页的元数据。每个元数据是一个结构体,包括用于记录数据页是否发生写访问的页面干净状态 DIRTY、冷热标志 COOL、数据页写入内存前的逻辑地址、数据页在缓冲链表中的前后指针 prev 和 next 等。考虑用于存储缓冲链表的高速缓存 (例如内存) 空间开销宝贵,因此本实施例中只维持一个有限长度的克隆链表 L3,且不直接保存整个被置换的数据页,而且仅仅保存被置换数据页的元数据。克隆链表 L3 中一个数据页的元数据只有 20bytes 的信息,然而每个数据页的大小一般都为 2KB 或 4KB,因此通过数据页的元数据维持克隆链表 L3 的开销很小。若克隆列表中包含了大量的干净页 (冷干净页和热干净页),说明当前主要访问类型为读访问,若克隆列表中包含了大量的脏页 (冷脏页和热脏页),说明当前主要访问类型为写访问。本实施例中读密集阈值和写密集阈值均取值为 70%,当克隆列表中冷干净页和热干净页数量之和占克隆列表中数据页总数的比例超过 70% 时则减少查找窗口的窗口大小,当克隆列表中冷脏页和热脏页数量之和占克隆列表中数据页总数量的比例超过 70% 时增加查找窗口的窗口大小,若当前访问模式位于读密集阈值和写密集阈值之间的状态时,即对缓冲链表的访问没有表现出特别明显的读密集或者写密集,则保持窗口大小不变。

[0070] 本实施例中,减少窗口大小后查找窗口的窗口大小 W 为 $W = (1-1/(\mu+1)) \cdot W_0$,增加窗口大小后查找窗口的窗口大小 W 为 $W = (1+1/(\mu+1)) \cdot W_0$,其中 W_0 为查找窗口的初始窗口大小, μ 为查找窗口的窗口大小 W 与缓冲链表总数据页数量的比值。本实施例中,过两个变量 p 、 q 来控制 W 的变化,取 $p = (1+1/(\mu+1))$, $q = (1-1/(\mu+1))$,每次调整查找窗口的窗口大小 W 前 p 、 q 都由当前的窗口大小 W 决定。其中, p 用于增量控制,每次增加窗口大小时令 $W = p \cdot W_0$,变化前的 W_0 越小则增量 p 越大,当 W 越大增量越小从而保持 W 变化的平滑性; q 用于减量控制,每次减小窗口大小时令 $W = q \cdot W_0$,变化前的 W_0 越大则减量 q 越大,查找窗口的调整更加合理,有利于充分发挥固态硬盘的特性。

[0071] 以上仅是本发明的优选实施方式,本发明的保护范围并不局限于上述实施例,凡属于本发明思路下的技术方案均属于本发明的保护范围。应当指出,对于本发明所属技术领域的普通技术人员来说,在不脱离本发明原理前提下的若干改进和润饰,这些改进和润饰也应视为本发明的保护范围。

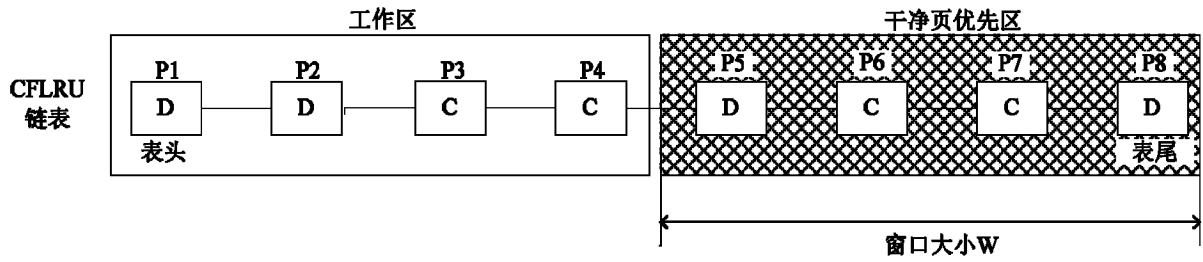


图 1

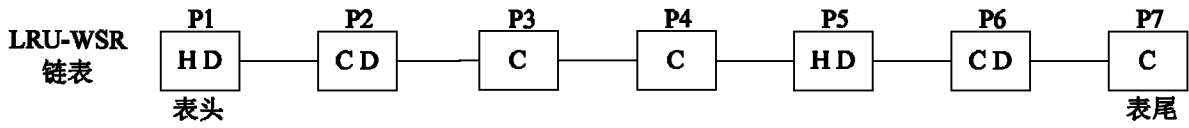


图 2

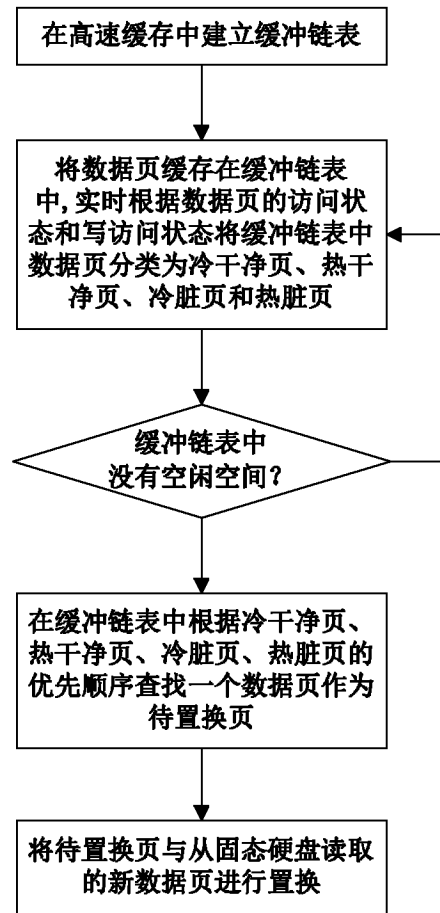


图 3

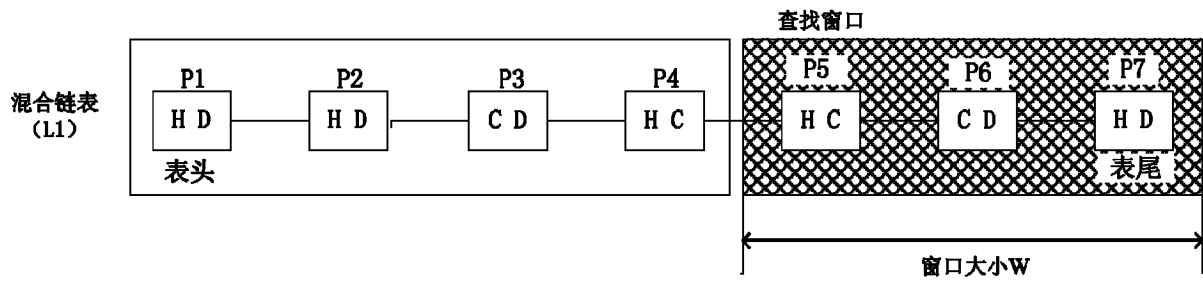


图 4

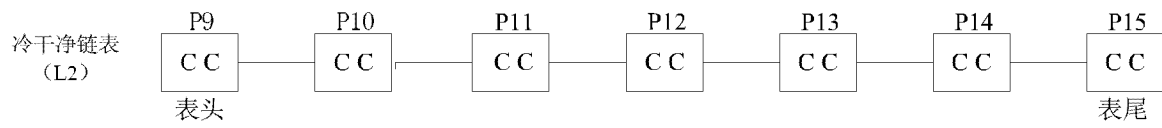


图 5

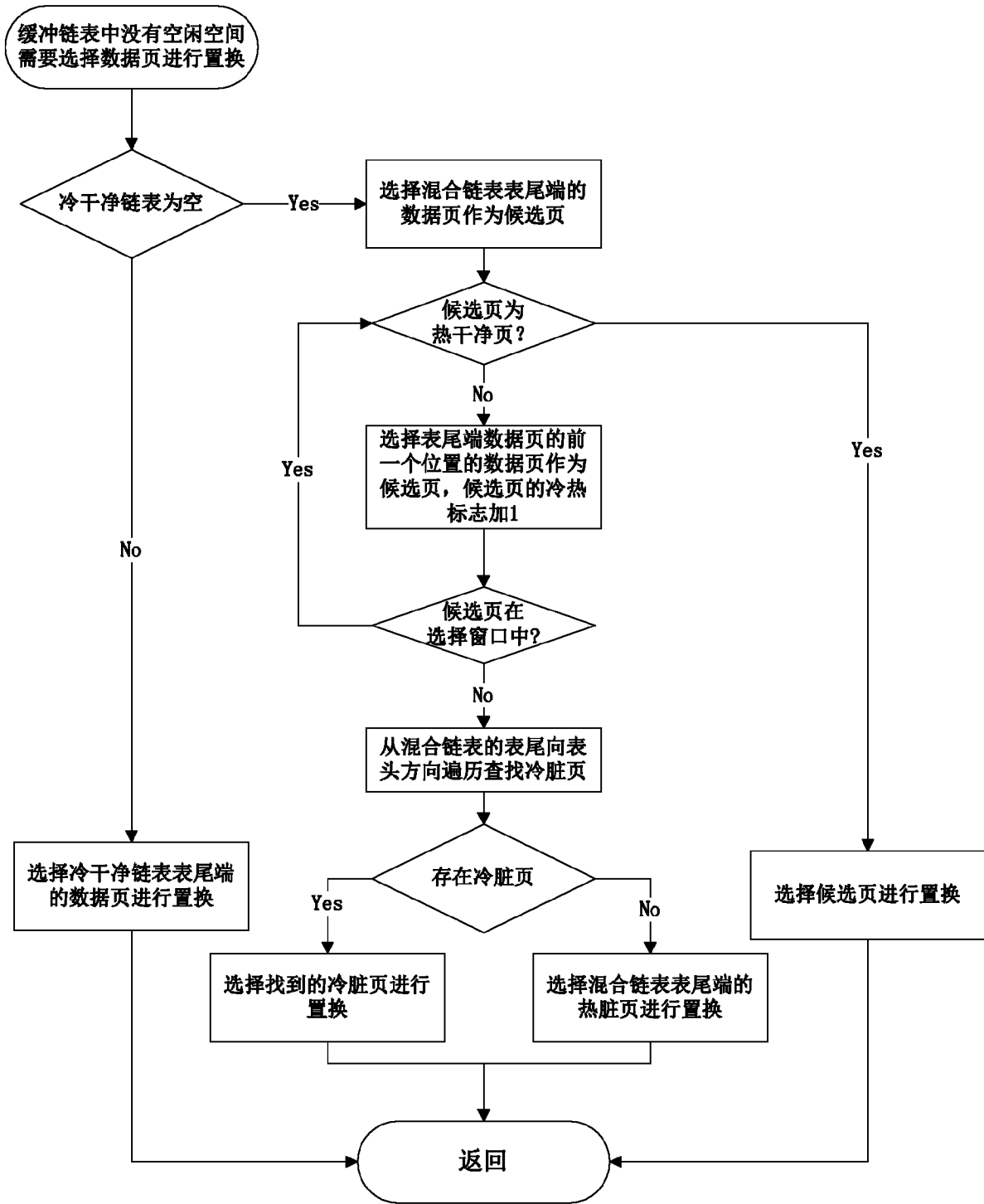


图 6

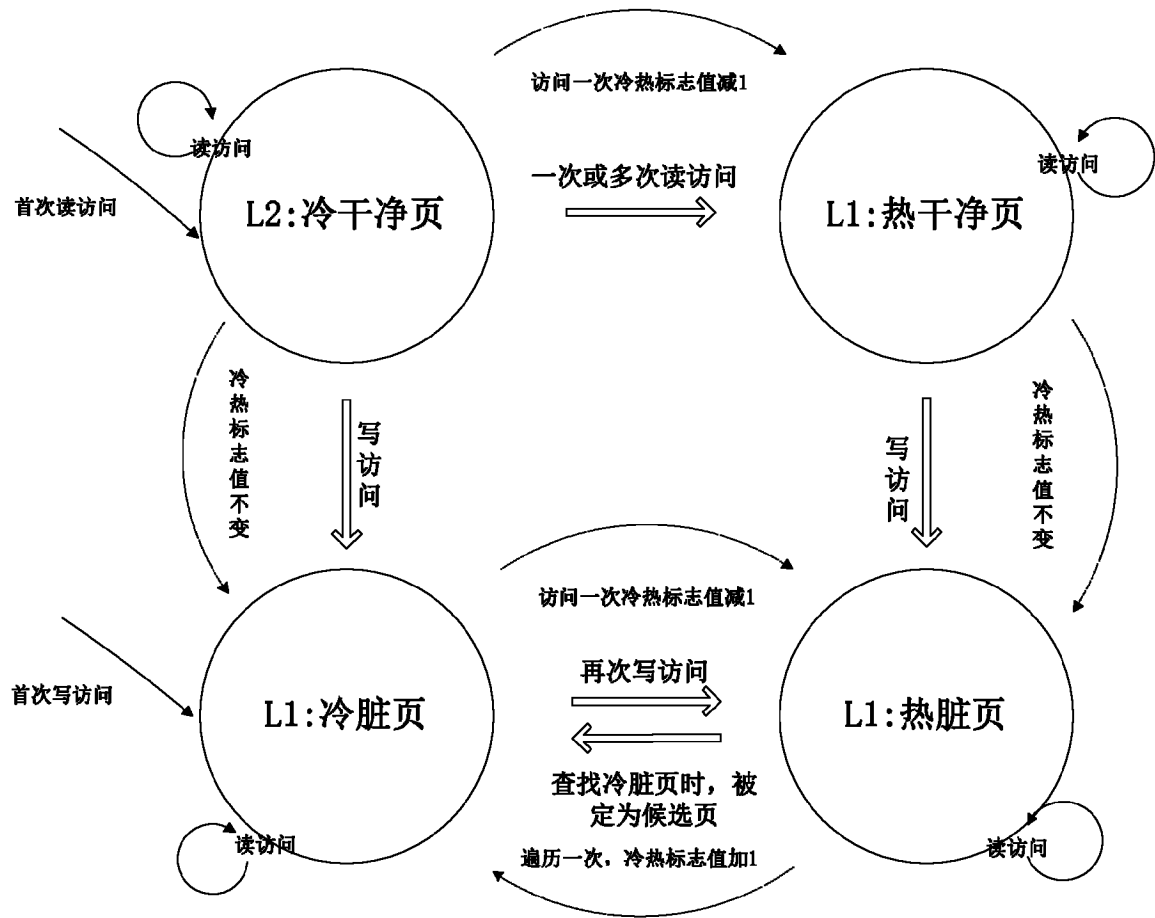


图 7

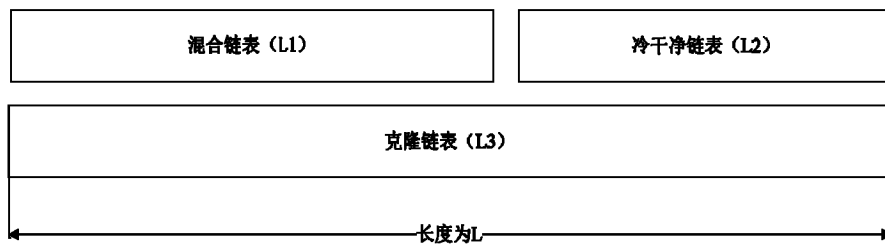


图 8