

(12) 按照专利合作条约所公布的国际申请

(19) 世界知识产权组织
国际局

(43) 国际公布日
2017年3月16日 (16.03.2017)



(10) 国际公布号
WO 2017/041673 A1

- (51) 国际专利分类号:
G06F 3/06 (2006.01)
- (21) 国际申请号: PCT/CN2016/098008
- (22) 国际申请日: 2016年9月5日 (05.09.2016)
- (25) 申请语言: 中文
- (26) 公布语言: 中文
- (30) 优先权:
201510572319.6 2015年9月10日 (10.09.2015) CN
- (71) 申请人: 阿里巴巴集团控股有限公司 (ALIBABA GROUP HOLDING LIMITED) [—/CN]; 开曼群岛英

属开曼群岛大开曼资本大厦一座四层 847 号邮箱, Grand Cayman (KY)。

(72) 发明人: 陶辉 (TAO, Hui); 中国浙江省杭州市余杭区文一西路 969 号 3 号楼 5 楼阿里巴巴集团法务部, Zhejiang 311121 (CN)。

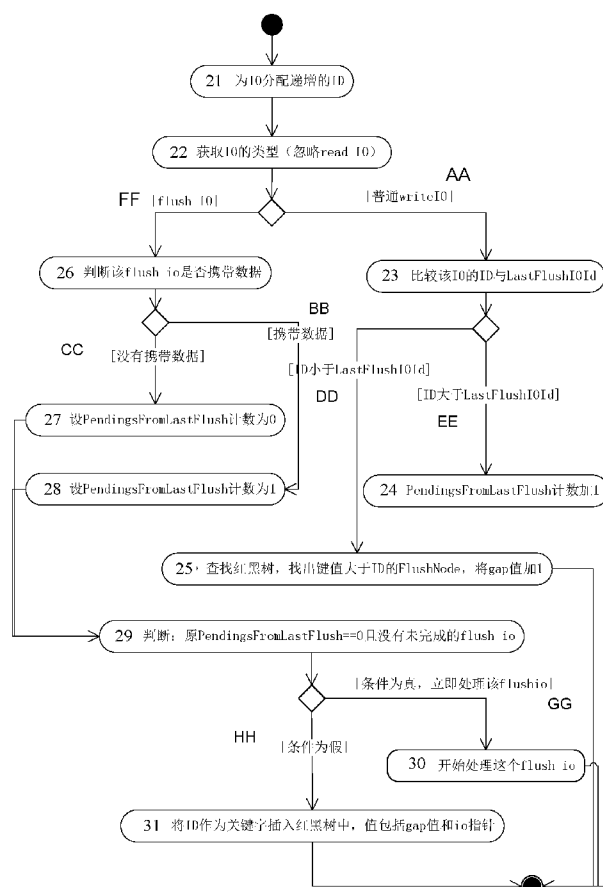
(74) 代理人: 北京三友知识产权代理有限公司 (BEIJING SANYOU INTELLECTUAL PROPERTY AGENCY LTD.); 中国北京市金融街 35 号国际企业大厦 A 座 16 层, Beijing 100033 (CN)。

(81) 指定国 (除另有指明, 要求每一种可提供的国家保护): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG,

[见续页]

(54) Title: PROCESSING METHOD AND DEVICE FOR FLUSH INPUT AND OUTPUT REQUEST OF MAGNETIC DISK

(54) 发明名称: 刷新磁盘输入输出请求的处理方法及设备



- 21 Allocating an incremental ID to an IO
22 Acquiring the type of the IO (neglecting a read IO)
23 Comparing the ID of the IO with LastFlushIOID
24 Incrementing a count of PendingFromLastFlush by 1
25 Searching for a red-black tree, finding a FlushNode of which a key value is greater than the ID, and incrementing a gap value by 1
26 Determining whether the flush IO carries data
27 Setting the count of the PendingFromLastFlush to 0
28 Setting the count of the PendingFromLastFlush to 1
29 Judgement: original PendingFromLastFlush == 0 and there is no unprocessed flush IO
30 Starting to process this flush IO
31 Using the ID as a keyword, and inserting same into the red-black tree, wherein the value comprises the gap value and an IO pointer
- AA Normal write IO
BB Carrying data
CC Not carrying data
DD ID is smaller than LastFlushIOID
EE ID is greater than LastFlushIOID
FF Flush IO
GG The condition is true: immediately processing the flush IO
HH The condition is false

图 6

[见续页]

WO 2017/041673 A1



BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW。

NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), 欧亚 (AM, AZ, BY, KG, KZ, RU, TJ, TM), 欧洲 (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG)。

本国际公布:

(84) 指定国 (除另有指明, 要求每一种可提供的地区保护): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ,

— 包括国际检索报告(条约第 21 条(3))。

(57) **Abstract:** A processing method and device for a flush magnetic disk input and output request. By determining whether a landed flush magnetic disk input and output request satisfies a pre-set condition and comparing an identification number of a landed magnetic disk input and output request with an identification number of a last flush magnetic disk input and output request, a flush magnetic disk input and output request can be rapidly processed according to a determination result, a CPU can be efficiently used, and in the case where a large amount of magnetic disk write input and output requests and flush magnetic disk input and output requests are concurrent in a cache existence scenario, the flush magnetic disk input and output requests and magnetic disk write input and output requests can still be efficiently processed, thereby realizing that a magnetic disk input and output request processing module efficiently processes flush magnetic disk input and output requests with low power consumption.

(57) **摘要:** 一种刷新磁盘输入输出请求的处理方法及设备, 通过判断所述到来的刷新磁盘输入输出请求是否满足预设条件, 及将落地的磁盘输入输出请求的标识号与上一个刷新磁盘输入输出请求的标识号进行比较, 可以根据判断结果快速处理刷新磁盘输入输出请求, 高效使用 CPU, 满足在缓存存在场景下大量的磁盘写输入输出请求与刷新磁盘输入输出请求并发时, 仍然能够高效的处理刷新磁盘输入输出请求与磁盘写输入输出请求, 实现磁盘输入输出请求处理模块高效率低功耗地处理刷新磁盘输入输出请求。

刷新磁盘输入输出请求的处理方法及设备

本申请要求 2015 年 09 月 10 日递交的申请号为 201510572319.6、发明名称为“刷新磁盘输入输出请求的处理方法及设备”的中国专利申请的优先权，其全部内容通过引用结合在本申请中。

5

技术领域

本申请涉及计算机领域，尤其涉及一种刷新磁盘输入输出请求的处理方法及设备。

背景技术

10 在虚拟化技术的（Xen）虚拟机环境的宿主机（domain0）中，当客户机虚拟机（domainU）发来的磁盘输入输出请求（IO）经过 domain0 的磁盘输入输出请求处理模块（如 TDC）继续处理时，需要能够正确处理刷新磁盘输入输出请求（Flush IO），处理方法为：当一个刷新磁盘输入输出请求来临时，之前未完成的磁盘写输入输出请求（write IO）必须全部完成后，才能够处理这个刷新磁盘输入输出请求。

15 刷新磁盘输入输出请求分为带数据与不带数据 2 种，处理刷新磁盘输入输出请求时，对带数据的刷新磁盘输入输出请求，需要将待写入的磁盘输入输出请求数据处理完，才能向虚拟机返回刷新磁盘输入输出请求处理完毕，对不带数据的刷新磁盘输入输出请求，则直接向虚拟机返回成功即可。

20 在一个刷新磁盘输入输出请求来临且未处理完成时，之后到来的磁盘写输入输出请求仍然能够继续处理。在一个刷新磁盘输入输出请求未处理完成时，允许更多的刷新磁盘输入输出请求继续到达模块。一个刷新磁盘输入输出请求开始处理的必要条件是：之前的所有磁盘写输入输出请求全部处理完成且返回虚拟机；之前的所有刷新磁盘输入输出请求全部处理完成且返回虚拟机。

25 若宿主机的磁盘输入输出请求处理模块支持缓存（cache），将会导致磁盘输入输出请求处理模块中会有数以万计的磁盘写输入输出请求和刷新磁盘输入输出请求都处于未完成状态（飞行状态）。

现有的刷新磁盘输入输出请求处理方式有如下 2 个明显的问题：

问题一：当刷新磁盘输入输出请求到达时，需要遍历包含全部的飞行中磁盘输入输出请求的链表（OngoingIOList），使用链表中的磁盘写输入输出请求个数以求得上一个

刷新磁盘输入输出请求与当前的刷新磁盘输入输出请求之间的磁盘写输入输出请求个数即累计间隙数值;

问题二: 当磁盘写输入输出请求完成时, 需要遍历包含全部刷新磁盘输入输出请求的链表 (FlushIOList), 以降低刷新磁盘输入输出请求的等待时间, 试图触发刷新磁盘输入输出请求的处理。

上述问题一要求飞行中的磁盘输入输出请求个数不能过大, 上述问题二要求刷新磁盘输入输出请求个数不能过大。一旦上述包含全部的飞行中磁盘输入输出请求的链表和包含全部刷新磁盘输入输出请求的链表数目开始从几十上升到数百、数千、数万时, 由于遍历链表这种 $O(N)$ 的算法复杂度, 将会导致最基础的磁盘输入输出请求处理流程损耗大量的 CPU 在计算累计间隙数值上, 而且处理时间还会增加。

所以, 上述现有的处理方式使得核心的磁盘输入输出请求流程在大量刷新磁盘输入输出请求开始出现时, 性能出现不必要的损耗, 尤其随着缓存技术的出现, 缓存回写模式 (writeback) 模式下缓存大量磁盘写输入输出请求更会导致性能的快速下降。

发明内容

本申请的一个目的是提供一种刷新磁盘输入输出请求的处理方法及设备, 能够解决在大量的磁盘写输入输出请求与刷新磁盘输入输出请求并发时, 处理刷新磁盘输入输出请求的效率下降的问题。

一种刷新磁盘输入输出请求的处理方法, 其中, 该方法包括:

在每一个磁盘输入输出请求到来时, 为每个磁盘输入输出请求递增地分配唯一的标识号, 其中, 所述磁盘输入输出请求包括磁盘写输入输出请求和刷新磁盘输入输出请求中的任一种;

判断所述到来的磁盘输入输出请求的种类,

若为刷新磁盘输入输出请求, 则判断所述到来的刷新磁盘输入输出请求是否满足预设条件, 根据判断结果确定是否处理所述到来的刷新磁盘输入输出请求。

进一步的, 上述方法中, 判断所述到来的刷新磁盘输入输出请求是否满足预设条件, 根据判断结果确定是否处理所述到来的刷新磁盘输入输出请求包括:

判断所述到来的刷新磁盘输入输出请求是否携带数据,

若携带数据, 将实时磁盘写输入输出请求的累计个数记录为原始累计个数后, 将实时磁盘写输入输出请求的累计个数更新为 1; 若没有携带数据, 将实时磁盘写输入输出

请求的累计个数记录为原始累计个数后，将实时磁盘写输入输出请求的累计个数更新为 0，其中，所述实时磁盘写输入输出请求的累计个数每次记录从上一个刷新磁盘输入输出请求至当前时刻所到达的磁盘写输入输出请求的累计个数；

根据所述原始累计个数确定是否处理所述到来的刷新磁盘输入输出请求。

- 5 进一步的，上述方法中，根据所述原始累计个数确定是否处理所述到来的刷新磁盘输入输出请求，包括：

判断所述原始累计个数是否为 0，且此时键码键值关联容器是否为空，其中，所述键码键值关联容器用于存放刷新磁盘输入输出请求的标识号和对应的节点信息，每个节点信息包括一个刷新磁盘输入输出请求和该刷新磁盘输入输出请求的累计间隙数，所述
10 累计间隙数的初始值为所述原始累计个数；

若所述原始累计个数为 0，且此时键码键值关联容器为空，立即处理该刷新磁盘输入输出请求；

若所述原始累计个数不为 0，或此时键码键值关联容器不为空，则将该到来的刷新磁盘输入输出请求的标识号和对应的节点信息插入所述键码键值关联容器中。

- 15 进一步的，上述方法中，所述键码键值关联容器为红黑树容器。

进一步的，上述方法中，判断所述到来的磁盘输入输出请求的种类之后，还包括：

若为磁盘写输入输出请求，将所述到来的磁盘写输入输出请求的标识号与上一个刷新磁盘输入输出请求的标识号比较，

- 若该磁盘写输入输出请求的标识号大于上一个刷新磁盘输入输出请求的标识号，则
20 将所述实时磁盘写输入输出请求的累计个数加 1；

若该磁盘写输入输出请求的标识号小于上一个刷新磁盘输入输出请求的标识号，则从所述键码键值关联容器中找到恰大于该磁盘写输入输出请求的标识号的刷新磁盘输入输出请求，并将该找到的刷新磁盘输入输出请求的累计间隙数加 1。

- 进一步的，上述方法中，根据判断结果确定是否处理所述到来的刷新磁盘输入输出
25 请求之后，还包括：

当一个磁盘输入输出请求的数据成功转发到实际存储介质中后，将该磁盘输入输出请求的标识号与上一个刷新磁盘输入输出请求的标识号进行比较，根据比较结果处理该磁盘输入输出请求，其中，磁盘输入输出请求包括磁盘写输入输出请求或带数据的刷新磁盘输入输出请求。

- 30 进一步的，上述方法中，将该磁盘输入输出请求的标识号与上一个刷新磁盘输入输

出请求的标识号进行比较, 根据比较结果处理该磁盘输入输出请求, 包括:

若该磁盘输入输出请求的标识号大于上一个刷新磁盘输入输出请求的标识号, 则将实时磁盘写输入输出请求的累计个数减 1。

进一步的, 上述方法中, 将该磁盘输入输出请求的标识号与上一个刷新磁盘输入输出请求的标识号进行比较, 根据比较结果处理该磁盘输入输出请求, 还包括:

若该磁盘输入输出请求的标识号小于等于上一个刷新磁盘输入输出请求的标识号, 则在所述键码键值关联容器中查找恰大于该磁盘输入输出请求的标识号的刷新磁盘输入输出请求, 并将该找到的刷新磁盘输入输出请求的累计间隙数减 1 后, 判断该找到的刷新磁盘输入输出请求的当前的累计间隙数是否为 0,

如果为 0, 判断该找到的刷新磁盘输入输出请求的标识号是否是所述键码键值关联容器中最小的标识号,

若是, 处理该找到的刷新磁盘输入输出请求后, 从键码键值关联容器中删除该找到的刷新磁盘输入输出请求的标识号和对应的节点信息;

重复如下步骤, 直至查找到的刷新磁盘输入输出请求的累计间隙数大于 0:

查找所述键码键值关联容器中标识号紧随于最近删除的刷新磁盘输入输出请求的标识号之后的刷新磁盘输入输出请求, 判断该查找到的刷新磁盘输入输出请求的累计间隙数是否也为 0, 若为 0, 则处理该查找到的刷新磁盘输入输出请求后, 从所述键码键值关联容器中删除该查找到的刷新磁盘输入输出请求的标识号和对应的节点信息。

根据本申请的一个方面, 提供了一种刷新磁盘输入输出请求的处理设备, 其中, 该设备包括:

分配标识号装置, 用于在每一个磁盘输入输出请求到来时, 为每个磁盘输入输出请求递增地分配唯一的标识号, 其中, 所述磁盘输入输出请求包括磁盘写输入输出请求和刷新磁盘输入输出请求中的任一种;

种类判断装置, 用于判断所述到来的磁盘输入输出请求的种类,

条件判断装置, 用于若为刷新磁盘输入输出请求, 则判断所述到来的刷新磁盘输入输出请求是否满足预设条件, 根据判断结果确定是否处理所述到来的刷新磁盘输入输出请求。

进一步的, 上述设备中, 所述条件判断装置, 用于判断所述到来的刷新磁盘输入输出请求是否携带数据,

若携带数据, 将实时磁盘写输入输出请求的累计个数记录为原始累计个数后, 将实

时磁盘写输入输出请求的累计个数更新为 1；若没有携带数据，将实时磁盘写输入输出请求的累计个数记录为原始累计个数后，将实时磁盘写输入输出请求的累计个数更新为 0，其中，所述实时磁盘写输入输出请求的累计个数每次记录从上一个刷新磁盘输入输出请求至当前时刻所到达的磁盘写输入输出请求的累计个数；

5 根据所述原始累计个数确定是否处理所述到来的刷新磁盘输入输出请求。

进一步的，上述设备中，所述条件判断装置，用于判断所述原始累计个数是否为 0，且此时键码键值关联容器是否为空，其中，所述键码键值关联容器用于存放刷新磁盘输入输出请求的标识号和对应的节点信息，每个节点信息包括一个刷新磁盘输入输出请求和该刷新磁盘输入输出请求的累计间隙数，所述累计间隙数的初始值为所述原始累计个数；

10 若所述原始累计个数为 0，且此时键码键值关联容器为空，立即处理该刷新磁盘输入输出请求；

若所述原始累计个数不为 0，或此时键码键值关联容器不为空，则将该到来的刷新磁盘输入输出请求的标识号和对应的节点信息插入所述键码键值关联容器中。

15 进一步的，上述设备中，所述键码键值关联容器为红黑树容器。

进一步的，上述设备中，所述设备还包括：

磁盘写输入输出请求处理装置，用于若为磁盘写输入输出请求，将所述到来的磁盘写输入输出请求的标识号与上一个刷新磁盘输入输出请求的标识号比较，

20 若该磁盘写输入输出请求的标识号大于上一个刷新磁盘输入输出请求的标识号，则将所述实时磁盘写输入输出请求的累计个数加 1；

若该磁盘写输入输出请求的标识号小于上一个刷新磁盘输入输出请求的标识号，则从所述键码键值关联容器中找到恰大于该磁盘写输入输出请求的标识号的刷新磁盘输入输出请求，并将该找到的刷新磁盘输入输出请求的累计间隙数加 1。

进一步的，上述设备中，所述设备还包括：

25 标识号比较装置，用于当一个磁盘输入输出请求的数据成功转发到实际存储介质中后，将该磁盘输入输出请求的标识号与上一个刷新磁盘输入输出请求的标识号进行比较，根据比较结果处理该磁盘输入输出请求，其中，磁盘输入输出请求包括磁盘写输入输出请求或带数据的刷新磁盘输入输出请求。

进一步的，上述设备中，所述标识号比较装置，用于若该磁盘输入输出请求的标识号大于上一个刷新磁盘输入输出请求的标识号，则将实时磁盘写输入输出请求的累计个

数减 1。

进一步的，上述设备中，所述标识号比较装置，还用于若该磁盘输入输出请求的标识号小于等于上一个刷新磁盘输入输出请求的标识号，则在所述键码键值关联容器中查找恰大于该磁盘输入输出请求的标识号的刷新磁盘输入输出请求，并将该找到的刷新磁
5 盘输入输出请求的累计间隙数减 1 后，判断该找到的刷新磁盘输入输出请求的当前的累计间隙数是否为 0，

如果为 0，判断该找到的刷新磁盘输入输出请求的标识号是否是所述键码键值关联容器中最小的标识号，

若是，处理该找到的刷新磁盘输入输出请求后，从键码键值关联容器中删除该找到
10 的刷新磁盘输入输出请求的标识号和对应的节点信息；

重复查找所述键码键值关联容器中标识号紧随于最近删除的刷新磁盘输入输出请求的标识号之后的刷新磁盘输入输出请求，判断该查找到的刷新磁盘输入输出请求的累计间隙数是否也为 0，若为 0，则处理该查找到的刷新磁盘输入输出请求后，从所述键码键值关联容器中删除该查找到的刷新磁盘输入输出请求的标识号和对应的节点信息，直至
15 查找到的刷新磁盘输入输出请求的累计间隙数大于 0。

与现有技术相比，本申请通过判断所述到来的刷新磁盘输入输出请求是否满足预设条件，可以根据判断结果能够快速处理刷新磁盘输入输出请求，高效使用 CPU，满足在缓存存在场景下大量的磁盘写输入输出请求与刷新磁盘输入输出请求并发时，仍然能够高效的处理刷新磁盘输入输出请求与 磁盘写输入输出请求，实现磁盘输入输出请求处理
20 模块高效率低功耗的处理刷新磁盘输入输出请求。

进一步的，本申请通过判断所述到来的刷新磁盘输入输出请求是否携带数据，能够根据判断结果更精确确定是否处理所述到来的刷新磁盘输入输出请求。

进一步的，本申请通过判断是否满足所述原始累计个数为 0，且此时键码键值关联容器为空，可以高效地判断地当前这个刷新磁盘输入输出请求是否可以立即执行。

进一步的，本申请通过将所述到来的磁盘写输入输出请求的标识号与上一个刷新磁盘输入输出请求的标识号比较，可以精确计算出所述到来的磁盘输入输出请求种类是磁盘写输入输出请求，对未来刷新磁盘输入输出请求的影响。
25

进一步的，本申请通过将磁盘输入输出请求的标识号与上一个刷新磁盘输入输出请求的标识号进行比较，能够精确判断出落地的磁盘输入输出请求是否会触发其他刷新磁
30 盘输入输出请求的完成，满足在缓存存在场景下大量的磁盘写输入输出请求与刷新磁盘

输入输出请求并发时，仍然能够高效的处理刷新磁盘输入输出请求与磁盘写输入输出请求，从而实现磁盘输入输出请求处理模块高效率低功耗的处理刷新磁盘输入输出请求。

进一步，本申请当磁盘写输入输出请求或者刷新磁盘输入输出请求完成时，通过在键码键值关联容器中查找恰大于该磁盘输入输出请求的标识号的刷新磁盘输入输出请求，可以 $O(\log N)$ 的算法复杂度处理，其中 N 是指刷新磁盘输入输出请求的数目，而不是指所有磁盘输入输出请求的数目，可以提高处理速度。

附图说明

通过阅读参照以下附图所作的对非限制性实施例所作的详细描述，本申请的其它特征、目的和优点将会变得更明显：

图 1 示出根据本申请一个方面的一种刷新磁盘输入输出请求的处理方法的流程图；

图 2 示出本申请一个优选实施例的刷新磁盘输入输出请求的处理方法流程图；

图 3 示出本申请另一个优选实施例的刷新磁盘输入输出请求的处理方法流程图；

图 4 示出本申请再一个优选实施例的刷新磁盘输入输出请求的处理方法流程图；

图 5 示出本申请又一个优选实施例的刷新磁盘输入输出请求的处理方法流程图；

图 6 示出本申请一具体的应用实施例的原理流程图；

图 7 示出本申请另一具体的应用实施例的原理流程图；

图 8 示出根据本申请另一个方面的一种刷新磁盘输入输出请求的处理设备的结构图；

图 9 示出本申请一个优选实施例的刷新磁盘输入输出请求的处理设备的结构图；

图 10 示出本申请另一个优选实施例的刷新磁盘输入输出请求的处理设备的结构图。

附图中相同或相似的附图标记代表相同或相似的部件。

具体实施方式

下面结合附图对本发明作进一步详细描述。

在本申请一个典型的配置中，终端、服务网络的设备和可信方均包括一个或多个处理器 (CPU)、输入/输出接口、网络接口和内存。

内存可能包括计算机可读介质中的非永久性存储器，随机存取存储器 (RAM) 和/或非易失性内存等形式，如只读存储器 (ROM) 或闪存 (flash RAM)。内存是计算机可读介质的示例。

计算机可读介质包括永久性和非永久性、可移动和非可移动媒体可以由任何方法或技术来实现信息存储。信息可以是计算机可读指令、数据结构、程序的模块或其他数据。计算机的存储介质的例子包括，但不限于相变内存 (PRAM)、静态随机存取存储器 (SRAM)、动态随机存取存储器 (DRAM)、其他类型的随机存取存储器 (RAM)、只读存储器 (ROM)、电可擦除可编程只读存储器 (EEPROM)、快闪记忆体或其他内存技术、只读光盘只读存储器 (CD-ROM)、数字多功能光盘 (DVD) 或其他光学存储、磁盒式磁带，磁带磁盘存储或其他磁性存储设备或任何其他非传输介质，可用于存储可以被计算设备访问的信息。按照本文中的界定，计算机可读介质不包括非暂存电脑可读媒体 (transitory media)，如调制的数据信号和载波。

根据本申请的一个方面，如图 1 所示，提供一种刷新磁盘输入输出请求的处理方法，包括：

步骤 S1，在每一个磁盘输入输出请求（磁盘输入输出请求）到来时，为每个磁盘输入输出请求递增地分配唯一的标识号，其中所述磁盘输入输出请求包括磁盘写输入输出请求（磁盘写输入输出请求）和刷新磁盘输入输出请求中的任一种；在此，通常磁盘输入输出请求包括磁盘写输入输出请求、磁盘读输入输出请求（read IO）和刷新磁盘输入输出请求，由于本申请中磁盘读输入输出请求与刷新磁盘输入输出请求处理无关，所以这里可以忽略磁盘读输入输出请求，所述刷新磁盘输入输出请求用于将缓存中的写数据最终存储到实际存储介质中；

步骤 S2，判断所述到来的磁盘输入输出请求的种类，

步骤 S3，若所述到来的磁盘输入输出请求种类是刷新磁盘输入输出请求，则判断所述到来的刷新磁盘输入输出请求是否满足预设条件，根据判断结果确定是否处理所述到来的刷新磁盘输入输出请求。在此，通过判断所述到来的刷新磁盘输入输出请求是否满足预设条件，可以根据判断结果能够快速处理刷新磁盘输入输出请求，高效使用 CPU，满足在缓存存在场景下大量的磁盘写输入输出请求与刷新磁盘输入输出请求并发时，仍然能够高效的处理刷新磁盘输入输出请求与 磁盘写输入输出请求，实现磁盘输入输出请求处理模块高效率低功耗的处理刷新磁盘输入输出请求。

本申请刷新磁盘输入输出请求的处理方法一优选的实施例中，如图 2 所示，步骤 S3，若所述到来的磁盘输入输出请求种类是刷新磁盘输入输出请求，则判断所述到来的刷新磁盘输入输出请求是否满足预设条件，根据判断结果确定是否处理所述到来的刷新磁盘输入输出请求，包括：

步骤 S31, 判断所述到来的刷新磁盘输入输出请求是否携带数据;

步骤 S32, 若携带数据, 将实时磁盘写输入输出请求的累计个数 (PendingsFromLastFlush) 记录为原始累计个数后, 并将实时磁盘写输入输出请求的累计个数更新为 1 后, 转到步骤 S34, 其中, 所述实时磁盘写输入输出请求的累计个数每次记录从上一个刷新磁盘输入输出请求至当前时刻所到达的磁盘写输入输出请求的累计个数; 在此, 由于磁盘输入输出请求处理模块 (如 TDC) 每收到 1 个磁盘输入输出请求都有独一无二的标识号, 这个标识号是可由 1 开始递增, 所以上一个刷新磁盘输入输出请求的标识号 (LastFlushIOId) 是个变量, 上一个刷新磁盘输入输出请求的标识号可初始化为 0, 即如果一直没有刷新磁盘输入输出请求, 则上一个刷新磁盘输入输出请求的标识号一直是 0;

步骤 S33, 若没有携带数据, 将实时磁盘写输入输出请求的累计个数记录为原始累计个数后, 将实时磁盘写输入输出请求的累计个数更新为 0 后, 转到步骤 S34; 在此, 将实时磁盘写输入输出请求的累计个数重新赋值为 1 表示当前的刷新磁盘输入输出请求也必须把数据下发成功;

步骤 S34, 根据所述原始累计个数确定是否处理所述到来的刷新磁盘输入输出请求, 从而根据所述原始累计个数能够更精确、快速处理刷新磁盘输入输出请求, 高效使用 CPU。

本申请刷新磁盘输入输出请求的处理方法一优选的实施例中, 如图 3 所示, 步骤 S34, 根据所述原始累计个数确定是否处理所述到来的刷新磁盘输入输出请求, 包括:

步骤 S341, 判断所述原始累计个数是否为 0, 且此时键码键值关联容器是否为空, 即没有未完成的刷新磁盘输入输出请求, 其中, 所述键码键值关联容器用于存放刷新磁盘输入输出请求的标识号和对应的节点信息, 每个节点信息包括该刷新磁盘输入输出请求和该刷新磁盘输入输出请求的累计间隙数 (gap), 所述累计间隙数的初始值为所述原始累计个数, 具体来说, 可在键码键值关联容器存放各个刷节点 (FlushNode), 每个刷节点用于描述一个刷新磁盘输入输出请求的标识号和对应的节点信息, 每个节点信息可包括 2 个成员: 累计间隙数和磁盘输入输出请求指针, 累计间隙数就是这个刷新磁盘输入输出请求与它前面的刷新磁盘输入输出请求之间, 还有多少个未完成的磁盘写输入输出请求, 而磁盘输入输出请求指针则指向所属的刷新磁盘输入输出请求;

步骤 S342, 若所述原始累计个数为 0, 且此时键码键值关联容器为空, 在此, 这 2 个条件用于判断当前这个刷新磁盘输入输出请求是否可以立即执行, 即这 2 个条件均为

真时，则立即处理该刷新磁盘输入输出请求，比如，如果未携带数据，直接返回执行成功，如果携带数据则立即下发，通过该 2 个条件的判断，可实现刷新磁盘输入输出请求到达时，以 $O(1)$ 的算法复杂度快速处理；

步骤 S343，若所述原始累计个数不为 0，或此时键码键值关联容器不为空，即 2 个
5 条件中只要有其中一个条件不满足，则将该到来的刷新磁盘输入输出请求的标识号和对
应的节点信息插入所述键码键值关联容器中。在此，需要在所述键码键值关联容器中构
造一个新的刷节点结构体，该刷节点的累计间隙数的原始值记为所述原始累计个数，即
步骤 S32、步骤 S33 中更新前的实时磁盘写输入输出请求的累计个数。具体可以设置好
该刷新磁盘输入输出请求的指针，以刷新磁盘输入输出请求的标识号作为键码，将刷节
10 点插入所述键码键值关联容器中后，流程处理完毕。本实施例中通过判断是否满足所述
原始累计个数为 0，且此时键码键值关联容器为空，可以高效地判断当前这个刷新磁盘
输入输出请求是否可以立即执行。

本申请刷新磁盘输入输出请求的处理方法一优选的实施例中，所述键码键值关联容
器为红黑树容器。在此，红黑树可作为一个存放刷新磁盘输入输出请求的键码-键值
15 (key-value) 关联容器，它的键码可以是刷新磁盘输入输出请求的标识号，而键值则是
刷节点。另外，所述键码键值关联容器还可替换为任何支持按元素键码大小遍历的、时
间复杂度在 $O(\log n)$ 以内的、可查询刚好大于给定键码元素的容器。

本申请刷新磁盘输入输出请求的处理方法一优选的实施例中，如图 4 所示，步骤 S2，
判断所述到来的磁盘输入输出请求种类，之后还包括：

20 步骤 S4，若所述到来的磁盘输入输出请求种类是磁盘写输入输出请求，则将所述到
来的磁盘写输入输出请求的标识号与上一个刷新磁盘输入输出请求的标识号比较，

步骤 S5，若该磁盘写输入输出请求的标识号大于上一个刷新磁盘输入输出请求的标
识号，则将实时磁盘写输入输出请求的累计个数加 1 后，刷新磁盘输入输出请求的相关
即逻辑处理完毕；

25 步骤 S6，若该磁盘写输入输出请求的标识号小于上一个刷新磁盘输入输出请求的标
识号，则从所述键码键值关联容器中找到恰大于该磁盘写输入输出请求的标识号的刷新
磁盘输入输出请求，并将该刷新磁盘输入输出请求的累计间隙数加 1。在此，不存在该
磁盘写输入输出请求的标识号等于上一个刷新磁盘输入输出请求的标识号的场景。正常
情况下，不会有该磁盘写输入输出请求的标识号小于上一个刷新磁盘输入输出请求的标
30 识号步骤的执行，只有多磁盘输入输出请求的写输入输出请求和刷新磁盘输入输出请求

并发执行时，才有可能走到本步骤。例如，可从红黑树中找到大于磁盘写输入输出请求的标识号的刷节点，将其累计间隙数值加 1 后，刷新磁盘输入输出请求相关逻辑即处理完毕。在此，通过将所述到来的磁盘写输入输出请求的标识号与上一个刷新磁盘输入输出请求的标识号比较，可以精确计算出所述到来的磁盘输入输出请求种类是磁盘写输入输出请求，对未来刷新磁盘输入输出请求的影响。

当一个磁盘输入输出请求的数据落地后（由 TDC 模块成功转发到实际存储介质中），可称为磁盘输入输出请求处理完成，这里包括磁盘读输入输出请求、磁盘写输入输出请求、带数据的刷新磁盘输入输出请求。其中，当磁盘写输入输出请求、带数据的刷新磁盘输入输出请求完成时，需要判断是否触发到刷新磁盘输入输出请求的处理，本申请刷新磁盘输入输出请求的处理方法一优选的实施例，根据判断结果确定是否处理所述到来的刷新磁盘输入输出请求之后，还包括：

步骤 S7，当一个磁盘输入输出请求的数据成功转发到实际存储介质中（落地）后，将该磁盘输入输出请求的标识号与上一个刷新磁盘输入输出请求的标识号进行比较，根据比较结果处理该磁盘输入输出请求，其中，磁盘输入输出请求包括磁盘写输入输出请求或带数据的刷新磁盘输入输出请求。在此，本实施例通过将磁盘输入输出请求的标识号与上一个刷新磁盘输入输出请求的标识号进行比较，能够精确判断出落地的磁盘输入输出请求是否会触发其他刷新磁盘输入输出请求的完成，满足在缓存存在场景下大量的磁盘写输入输出请求与刷新磁盘输入输出请求并发时，仍然能够高效的处理刷新磁盘输入输出请求与 磁盘写输入输出请求，从而实现磁盘输入输出请求处理模块高效率低功耗的处理刷新磁盘输入输出请求。

本申请刷新磁盘输入输出请求的处理方法一优选的实施例中，步骤 S7，将该磁盘输入输出请求的标识号与上一个刷新磁盘输入输出请求的标识号进行比较，根据比较结果处理该磁盘输入输出请求，包括：

步骤 S71，若磁盘输入输出请求的标识号大于上一个刷新磁盘输入输出请求的标识号，则将实时磁盘写输入输出请求的累计个数减 1 后，从而为后续的刷新磁盘输入输出请求触发流程作好准备。在此，磁盘输入输出请求的标识号大于上一个刷新磁盘输入输出请求的标识号，则说明这个磁盘输入输出请求是磁盘写输入输出请求。

本申请刷新磁盘输入输出请求的处理方法一优选的实施例中，如图 5 所示，步骤 S7，将该磁盘输入输出请求的标识号与上一个刷新磁盘输入输出请求的标识号进行比较，根据比较结果处理该磁盘输入输出请求，还包括：

步骤 S72, 若该磁盘输入输出请求的标识号小于等于上一个刷新磁盘输入输出请求的标识号, 则在键码键值关联容器中查找恰大于该磁盘输入输出请求的标识号的刷新磁盘输入输出请求, 并将该找到的刷新磁盘输入输出请求的累计间隙数减 1 后, 判断该找到的刷新磁盘输入输出请求的当前的累计间隙数是否为 0, 如果为 0, 转到步骤 S73, 如果不为 0, 转到步骤 S77, 在此, 若该磁盘输入输出请求的标识号小于上一个刷新磁盘输入输出请求的标识号, 则说明该磁盘输入输出请求为磁盘写输入输出请求, 若该磁盘输入输出请求的标识号等于上一个刷新磁盘输入输出请求的标识号, 则说明该磁盘输入输出请求为带数据的刷新磁盘输入输出请求, 磁盘写输入输出请求或者刷新磁盘输入输出请求完成时, 通过在键码键值关联容器中查找恰大于该磁盘输入输出请求的标识号的刷新磁盘输入输出请求, 可以 $O(\log N)$ 的算法复杂度处理, 其中 N 是指刷新磁盘输入输出请求的数目, 而不是指所有磁盘输入输出请求的数目, 所以处理速度很快;

步骤 S73, 判断该找到的刷新磁盘输入输出请求的标识号是否是所述键码键值关联容器中最小的标识号, 若是, 转到步骤 S74, 若不是, 转到步骤 S76,

步骤 S74, 这个刷新磁盘输入输出请求就是所有未完成的刷新磁盘输入输出请求中的第 1 个磁盘输入输出请求, 处理该找到的刷新磁盘输入输出请求后, 从键码键值关联容器中删除该找到的刷新磁盘输入输出请求的标识号和对应的节点信息, 在此, 可以正常处理该刷新磁盘输入输出请求, 例如, 如果不携带数据, 则直接向虚拟机返回成功; 如果携带数据, 则将数据下发到存储介质落地, 既然已经处理完毕, 就从键码键值关联容器中删除这个刷节点;

步骤 S75, 重复如下步骤, 直至紧随于最近删除的刷新磁盘输入输出请求的标识号之后的刷新磁盘输入输出请求的累计间隙数大于 0, 流程结束:

查找所述键码键值关联容器中标识号紧随于最近删除的刷新磁盘输入输出请求, 判断查找到的刷新磁盘输入输出请求的标识号之后的刷新磁盘输入输出请求的累计间隙数是否也为 0, 也满足了触发条件, 若为 0, 则处理该找到的刷新磁盘输入输出请求后, 从键码键值关联容器中删除该找到的刷新磁盘输入输出请求的标识号和对应的节点信息;

步骤 S76, 如果该找到的刷新磁盘输入输出请求的标识号不是所述键码键值关联容器中最小的标识号, 则说明这个磁盘输入输出请求虽然导致它之后的刷新磁盘输入输出请求的累计间隙数值降为 0, 但是由于这个刷新磁盘输入输出请求前还有未完成的刷新磁盘输入输出请求, 所以刷新磁盘输入输出请求并不能执行;

步骤 S77, 若找到的刷新磁盘输入输出请求的标识号之后的刷新磁盘输入输出请求

的累计间隙数不为 0，则没有触发其他刷新磁盘输入输出请求的执行，流程结束。

详细的，本申请总体流程包含两部分：磁盘输入输出请求到达 TDC 后 flush 相关逻辑的处理流程，如果是写磁盘输入输出请求需要计算出对未来刷新磁盘输入输出请求的影响，如果是刷新磁盘输入输出请求则需要计算什么时候才可以处理它；磁盘输入输出请求数据落地后的处理流程，即磁盘写输入输出请求和带数据的刷新磁盘输入输出请求完成时，是否会触发其他刷新磁盘输入输出请求的完成。

本发明一具体应用实施例中，磁盘输入输出请求到达处理流程如图 6 所示，包括如下步骤：

步骤 21，每一个磁盘输入输出请求到来时，递增的分配标识号，包括磁盘读输入输出请求、磁盘写输入输出请求、刷新磁盘输入输出请求；

步骤 22，如果这个磁盘输入输出请求是磁盘写输入输出请求，则跳到第 23 步执行；如果是刷新磁盘输入输出请求，则跳到第 26 步执行；如果是磁盘读输入输出请求，因为与刷新磁盘输入输出请求处理无关，忽略；

步骤 23，将第 21 步分配到的标识号与上一个刷新磁盘输入输出请求的标识号比较，如果该标识号大于上一个刷新磁盘输入输出请求的标识号，则跳到第 24 步执行；如果小于上一个刷新磁盘输入输出请求的标识号，则跳到第 25 步执行，不存在等于的场景；

步骤 24，将实时磁盘写输入输出请求的累计个数计数加 1，刷新磁盘输入输出请求相关逻辑处理完毕；

步骤 25，正常情况下不会有这个步骤的执行，只有多磁盘输入输出请求的第 1 步与第 11 步并发执行时，才有可能走到本步骤。从红黑树中找到大于请求 ID 的刷节点（可对应于 c++ stl 的 map 的 upper_bound 方法），将其累计间隙数值加 1。刷新磁盘输入输出请求相关逻辑处理完毕；

步骤 26，判断这个刷新磁盘输入输出请求是否携带数据，如果携带数据，则跳到第 28 步执行；如果没有携带数据，则继续执行第 27 步；

步骤 27，将实时磁盘写输入输出请求的累计个数这个计数变量清 0；

步骤 28，将实时磁盘写输入输出请求的累计个数这个计数变量设置为 1，这个 1 表示当前的刷新磁盘输入输出请求也必须把数据下发成功；

步骤 29，在第 27、28 步执行前，原先的实时磁盘写输入输出请求的累计个数计数判断是否为 0，且没有未完成的刷新磁盘输入输出请求（判断存放刷新磁盘输入输出请求的红黑树容器是否为空即可），这一步是用于判断当前这个刷新磁盘输入输出请求是

否可以立即执行，如果上述 2 个条件为真，则继续执行第 30 步；否则，跳到 31 步执行；

步骤 30，立即处理这个刷新磁盘输入输出请求，比如，如果未携带数据，直接返回执行成功；如果携带数据则立即下发。

步骤 31，构造刷节点结构体，累计间隙数值记为原实时磁盘写输入输出请求的累计个数（第 27、28 步执行前的原实时磁盘写输入输出请求的累计个数），同时设置好该刷新磁盘输入输出请求的指针，以刷新磁盘输入输出请求的标识号作为键码，将刷节点插入红黑树容器中，流程处理完毕。

另外，当一个磁盘输入输出请求的数据落地后（由 TDC 模块成功转发到实际存储介质中），可称为磁盘输入输出请求处理完成，这里包括读磁盘输入输出请求、写磁盘输入输出请求、带数据的刷新磁盘输入输出请求。其中，当写磁盘输入输出请求、带数据的刷新磁盘输入输出请求完成时，需要判断是否触发到刷新磁盘输入输出请求的处理，如图 7 所示，当一个磁盘输入输出请求的数据落地后，后续的判断是否触发到刷新磁盘输入输出请求的处理步骤包括：

步骤 41，当一个写磁盘输入输出请求或者带数据的刷新磁盘输入输出请求落地后，首先将磁盘输入输出请求的标识号与上一个刷新磁盘输入输出请求的标识号比较，如果标识号小于等于上一个刷新磁盘输入输出请求的标识号，则跳到第 3 步执行；如果标识号大于上一个刷新磁盘输入输出请求的标识号，则继续执行第 42 步。

步骤 42，将实时磁盘写输入输出请求的累计个数计数减 1，刷新磁盘输入输出请求触发流程处理完毕。

步骤 43，在红黑树容器中查找恰大于该磁盘输入输出请求的标识号的刷节点结点（可对应于 c++ stl 的 map 的 upper_bound 方法）。

步骤 44，将第 43 步找到的刷节点结点的累计间隙数计数值减去 1，再判断累计间隙数值是否为 0，如果为 0，继续执行第 45 步；如果大于 0，则没有触发其他刷新磁盘输入输出请求的执行，流程结束；

步骤 45，检查当前的刷节点是否为红黑树中 ID 最小的结点（可中序遍历的第 1 个结点，对应于 c++ 的 stl 库中 map 的 begin 方法），如果不是则说明这个磁盘输入输出请求虽然导致它之后的刷新磁盘输入输出请求的累计间隙数值降为 0，但是由于这个刷新磁盘输入输出请求前还有未完成的刷新磁盘输入输出请求，所以刷新磁盘输入输出请求并不能执行；反之，这个刷新磁盘输入输出请求就是所有未完成的刷新磁盘输入输出请求中的第 1 个磁盘输入输出请求，继续执行第 46 步处理。

步骤 46,正常处理该刷新磁盘输入输出请求,例如,如果不携带数据,则直接向虚拟机返回成功;如果携带数据,则将数据下发到存储介质落地。

步骤 47,既然已经处理完毕,就从红黑树中删除这个刷节点;

步骤 48,接着需要查看红黑树中紧随其后的刷新磁盘输入输出请求(可对应于 c++ 的 stl 库中 map 的 ++ 重载方法)中是否累计间隙数值也为 0,也满足了触发条件,如果累计间隙数为 0,则继续执行第 46 步;否则,流程结束。

根据本申请的另一面,还提供一种刷新磁盘输入输出请求的处理设备,如图 8 所示,该设备 100 包括:

分配标识号装置 1,用于在每一个磁盘输入输出请求到来时,为每个磁盘输入输出请求递增地分配唯一的标识号,其中所述磁盘输入输出请求包括磁盘写输入输出请求和刷新磁盘输入输出请求(刷新磁盘输入输出请求)中的任一种;在此,通常磁盘输入输出请求包括磁盘写输入输出请求、磁盘读输入输出请求和刷新磁盘输入输出请求,由于本申请中磁盘读输入输出请求与刷新磁盘输入输出请求处理无关,所以这里可以忽略磁盘读输入输出请求,所述刷新磁盘输入输出请求用于将缓存中的写数据最终存储到实际存储介质中;

种类判断装置 2,用于判断所述到来的磁盘输入输出请求的种类,

条件判断装置 3,用于若所述到来的磁盘输入输出请求种类是刷新磁盘输入输出请求,则判断所述到来的刷新磁盘输入输出请求是否满足预设条件,根据判断结果确定是否处理所述到来的刷新磁盘输入输出请求。在此,通过判断所述到来的刷新磁盘输入输出请求是否满足预设条件,可以根据判断结果能够快速处理刷新磁盘输入输出请求,高效使用 CPU,满足在缓存存在场景下大量的磁盘写输入输出请求与刷新磁盘输入输出请求并发时,仍然能够高效的处理刷新磁盘输入输出请求与 磁盘写输入输出请求,实现磁盘输入输出请求处理模块高效率低功耗的处理刷新磁盘输入输出请求。

本申请刷新磁盘输入输出请求的处理设备一优选的实施例中,所述条件判断装置 3,用于判断所述到来的刷新磁盘输入输出请求是否携带数据,

若携带数据,将实时磁盘写输入输出请求的累计个数记录为原始累计个数后,将实时磁盘写输入输出请求的累计个数(实时磁盘写输入输出请求的累计个数)更新为 1,其中,所述实时磁盘写输入输出请求的累计个数每次记录从上一个刷新磁盘输入输出请求 (LastFlushIOId) 至当前时刻所到达的磁盘写输入输出请求的累计个数;在此,由于磁盘输入输出请求处理模块(如 TDC)每收到 1 个磁盘输入输出请求都有独一无二的标

识号，这个标识号是可由 1 开始递增，所以上一个刷新磁盘输入输出请求的标识号是个变量，上一个刷新磁盘输入输出请求的标识号可初始化为 0，即如果一直没有刷新磁盘输入输出请求，则上一个刷新磁盘输入输出请求的标识号一直是 0；若没有携带数据，将实时磁盘写输入输出请求的累计个数记录为原始累计个数后，将实时磁盘写输入输出请求的累计个数更新为 0，其中，所述实时磁盘写输入输出请求的累计个数每次记录从上一个刷新磁盘输入输出请求至当前时刻所到达的磁盘写输入输出请求的累计个数，在此，将实时磁盘写输入输出请求的累计个数重新赋值为 1 表示当前的刷新磁盘输入输出请求也必须把数据下发成功；

根据所述原始累计个数确定是否处理所述到来的刷新磁盘输入输出请求，从而根据所述原始累计个数能够更精确、快速处理刷新磁盘输入输出请求，高效使用 CPU。

本申请刷新磁盘输入输出请求的处理设备一优选的实施例中，所述条件判断装置 3，用于判断所述原始累计个数是否为 0，且此时键码键值关联容器是否为空，即没有未完成的刷新磁盘输入输出请求，其中，所述键码键值关联容器用于存放刷新磁盘输入输出请求的标识号和对应的节点信息，每个节点信息包括一个刷新磁盘输入输出请求和该刷新磁盘输入输出请求的累计间隙数（gap），所述累计间隙数的初始值为所述原始累计个数，具体来说，可在键码键值关联容器存放各个刷节点（FlushNode），每个刷节点用于描述一个刷新磁盘输入输出请求的标识号和对应的节点信息，每个节点信息可包括 2 个成员：累计间隙数和磁盘输入输出请求指针，累计间隙数就是这个刷新磁盘输入输出请求与它前面的刷新磁盘输入输出请求之间，还有多少个未完成的磁盘写输入输出请求，而磁盘输入输出请求指针则指向所属的刷新磁盘输入输出请求；

若所述原始累计个数为 0，且此时键码键值关联容器为空在此，这 2 个条件用于判断当前这个刷新磁盘输入输出请求是否可以立即执行，即这 2 个条件均为真时，则立即处理该刷新磁盘输入输出请求，比如，如果未携带数据，直接返回执行成功，如果携带数据则立即下发，通过该 2 个条件的判断，可实现刷新磁盘输入输出请求到达时，以 $O(1)$ 的算法复杂度快速处理；

若所述原始累计个数不为 0，或此时键码键值关联容器不为空，即 2 个条件中只要有其中一个条件不满足，则将该到来的刷新磁盘输入输出请求的标识号和对应的节点信息插入所述键码键值关联容器中。在此，需要在所述键码键值关联容器中构造一个新的刷节点结构体，该刷节点的累计间隙数的原始值记为所述原始累计个数，即更新前的实时磁盘写输入输出请求的累计个数（PendingsFromLastFlush）。具体可以设置好该刷新

磁盘输入输出请求的指针，以刷新磁盘输入输出请求的标识号作为键码，将刷节点插入所述键码键值关联容器中后，流程处理完毕。本实施例中通过判断是否满足所述原始累计个数为 0，且此时键码键值关联容器为空，可以高效地判断当前这个刷新磁盘输入输出请求是否可以立即执行。

5 本申请刷新磁盘输入输出请求的处理设备一优选的实施例中，所述键码键值关联容器为红黑树容器。在此，红黑树可作为一个存放刷新磁盘输入输出请求的键码-键值（key-value）关联容器，它的键码可以是刷新磁盘输入输出请求的标识号，而键值则是刷节点。另外，所述键码键值关联容器还可替换为任何支持按元素键码大小遍历的、时间复杂度在 $O(\log n)$ 以内的、可查询刚好大于给定键码的容器。

10 本申请刷新磁盘输入输出请求的处理设备一优选的实施例中，如图 9 所示，所述设备 100 还包括：

 磁盘写输入输出请求处理装置 4，用于若为磁盘写输入输出请求，将所述到来的磁盘写输入输出请求的标识号与上一个刷新磁盘输入输出请求的标识号比较，

 若该磁盘写输入输出请求的标识号大于上一个刷新磁盘输入输出请求的标识号，则将所述实时磁盘写输入输出请求的累计个数加 1，刷新磁盘输入输出请求的相关即逻辑处理完毕；

 若该磁盘写输入输出请求的标识号小于上一个刷新磁盘输入输出请求的标识号，则从所述键码键值关联容器中找到恰大于该磁盘写输入输出请求的标识号的刷新磁盘输入输出请求，并将该找到的刷新磁盘输入输出请求的累计间隙数（gap）加 1。在此，不存在该磁盘写输入输出请求的标识号等于上一个刷新磁盘输入输出请求的标识号的场景。正常情况下，不会有该磁盘写输入输出请求的标识号小于上一个刷新磁盘输入输出请求的标识号步骤的执行，只有多磁盘输入输出请求的写输入输出请求和刷新磁盘输入输出请求并发执行时，才有可能走到本步骤。例如，可从红黑树中找到大于磁盘写输入输出请求的标识号的刷节点，将其累计间隙数值加 1 后，刷新磁盘输入输出请求相关逻辑即

20 处理完毕。在此，通过将所述到来的磁盘写输入输出请求的标识号与上一个刷新磁盘输入输出请求的标识号比较，可以精确计算出所述到来的磁盘输入输出请求种类是磁盘写输入输出请求，对未来刷新磁盘输入输出请求的影响。

 当一个磁盘输入输出请求的数据落地后（由 TDC 模块成功转发到实际存储介质中），可称为磁盘输入输出请求处理完成，这里包括磁盘读输入输出请求、磁盘写输入输出请求、带数据的刷新磁盘输入输出请求。其中，当磁盘写输入输出请求、带数据的刷新磁

30

盘输入输出请求完成时，需要判断是否触发到刷新磁盘输入输出请求的处理，本申请刷新磁盘输入输出请求的处理设备一优选的实施例中，如图 10 所示，所述设备 100 还包括：

标识号比较装置 5，用于当一个磁盘输入输出请求的数据成功转发到实际存储介质中后，将该磁盘输入输出请求的标识号与上一个刷新磁盘输入输出请求的标识号进行比较，根据比较结果处理该磁盘输入输出请求，其中，磁盘输入输出请求包括磁盘写输入输出请求或带数据的刷新磁盘输入输出请求。在此，本实施例通过将磁盘输入输出请求的标识号与上一个刷新磁盘输入输出请求的标识号进行比较，能够精确判断出落地的磁盘输入输出请求是否会触发其他刷新磁盘输入输出请求的完成，满足在缓存存在场景下大量的磁盘写输入输出请求与刷新磁盘输入输出请求并发时，仍然能够高效的处理刷新
10 磁盘输入输出请求与 磁盘写输入输出请求，从而实现磁盘输入输出请求处理模块高效率低功耗的处理刷新磁盘输入输出请求。

本申请刷新磁盘输入输出请求的处理设备一优选的实施例中，所述标识号比较装置 5，用于若该磁盘输入输出请求的标识号大于上一个刷新磁盘输入输出请求的标识号，则将实时磁盘写输入输出请求的累计个数减 1，从而为后续的刷新磁盘输入输出请求触发
15 流程作好准备。在此，磁盘输入输出请求的标识号大于上一个刷新磁盘输入输出请求的标识号，则说明这个磁盘输入输出请求是磁盘写输入输出请求。

本申请刷新磁盘输入输出请求的处理设备一优选的实施例中，所述标识号比较装置 5，还用于若该磁盘输入输出请求的标识号小于等于上一个刷新磁盘输入输出请求的标识号，则在所述键码键值关联容器中查找恰大于该磁盘输入输出请求的标识号的刷新磁盘
20 输入输出请求，并将该找到的刷新磁盘输入输出请求的累计间隙数减 1 后，判断该找到的刷新磁盘输入输出请求的当前的累计间隙数是否为 0，在此，若该磁盘输入输出请求的标识号小于上一个刷新磁盘输入输出请求的标识号，则说明该磁盘输入输出请求为磁盘写输入输出请求，若该磁盘输入输出请求的标识号等于上一个刷新磁盘输入输出请求的标识号，则说明该磁盘输入输出请求为带数据的刷新磁盘输入输出请求， 磁盘写输入
25 输出请求或者刷新磁盘输入输出请求完成时，通过在键码键值关联容器中查找恰大于该磁盘输入输出请求的标识号的刷新磁盘输入输出请求，可以 $O(\log N)$ 的算法复杂度处理，其中 N 是指刷新磁盘输入输出请求的数目，而不是指所有磁盘输入输出请求的数目，所以处理速度很快；

如果为 0，判断该找到的刷新磁盘输入输出请求的标识号是否是所述键码键值关联
30 容器中最小的标识号，

若是，这个刷新磁盘输入输出请求就是所有未完成的刷新磁盘输入输出请求中的第 1 个磁盘输入输出请求，处理该找到的刷新磁盘输入输出请求后，从键码键值关联容器中删除该找到的刷新磁盘输入输出请求的标识号和对应的节点信息，在此，可以正常处理该刷新磁盘输入输出请求，例如，如果不携带数据，则直接向虚拟机返回成功；如果携带数据，则将数据下发到存储介质落地，既然已经处理完毕，就从键码键值关联容器中删除这个刷节点；

重复查找所述键码键值关联容器中标识号紧随于最近删除的刷新磁盘输入输出请求的标识号之后的刷新磁盘输入输出请求，判断该查找到的刷新磁盘输入输出请求的累计间隙数是否也为 0，若为 0，则处理该查找到的刷新磁盘输入输出请求后，从所述键码键值关联容器中删除该查找到的刷新磁盘输入输出请求的标识号和对应的节点信息，直至查找到的刷新磁盘输入输出请求的累计间隙数大于 0。

综上所述，本申请通过判断所述到来的刷新磁盘输入输出请求是否满足预设条件，可以根据判断结果能够快速处理刷新磁盘输入输出请求，高效使用 CPU，满足在缓存存在场景下大量的磁盘写输入输出请求与刷新磁盘输入输出请求并发时，仍然能够高效的处理刷新磁盘输入输出请求与 磁盘写输入输出请求，实现磁盘输入输出请求处理模块高效率低功耗的处理刷新磁盘输入输出请求。

进一步的，本申请通过判断所述到来的刷新磁盘输入输出请求是否携带数据，能够根据判断结果更精确确定是否处理所述到来的刷新磁盘输入输出请求。

进一步的，本申请通过判断是否满足所述原始累计个数为 0，且此时键码键值关联容器为空，可以高效地判断地当前这个刷新磁盘输入输出请求是否可以立即执行。

进一步的，本申请通过将所述到来的磁盘写输入输出请求的标识号与上一个刷新磁盘输入输出请求的标识号比较，可以精确计算出所述到来的磁盘输入输出请求种类是磁盘写输入输出请求，对未来刷新磁盘输入输出请求的影响。

进一步的，本申请通过将磁盘输入输出请求的标识号与上一个刷新磁盘输入输出请求的标识号进行比较，能够精确判断出落地的磁盘输入输出请求是否会触发其他刷新磁盘输入输出请求的完成，满足在缓存存在场景下大量的磁盘写输入输出请求与刷新磁盘输入输出请求并发时，仍然能够高效的处理刷新磁盘输入输出请求与 磁盘写输入输出请求，从而实现磁盘输入输出请求处理模块高效率低功耗的处理刷新磁盘输入输出请求。

进一步，本申请当磁盘写输入输出请求或者刷新磁盘输入输出请求完成时，通过在键码键值关联容器中查找恰大于该磁盘输入输出请求的标识号的刷新磁盘输入输出请

求，可以 $O(\log N)$ 的算法复杂度处理，其中 N 是指刷新磁盘输入输出请求的数目，而不是指所有磁盘输入输出请求的数目，可以提高处理速度。

显然，本领域的技术人员可以对本申请进行各种改动和变型而不脱离本申请的精神和范围。这样，倘若本申请的这些修改和变型属于本申请权利要求及其等同技术的范围之内，则本申请也意图包含这些改动和变型在内。

需要注意的是，本发明可在软件和/或软件与硬件的组合体中被实施，例如，可采用专用集成电路（ASIC）、通用目的计算机或任何其他类似硬件设备来实现。在一个实施例中，本发明的软件程序可以通过处理器执行以实现上文所述步骤或功能。同样地，本发明的软件程序（包括相关的数据结构）可以被存储到计算机可读记录介质中，例如，RAM 存储器，磁或光驱动器或软磁盘及类似设备。另外，本发明的一些步骤或功能可采用硬件来实现，例如，作为与处理器配合从而执行各个步骤或功能的电路。

另外，本发明的一部分可被应用为计算机程序产品，例如计算机程序指令，当其被计算机执行时，通过该计算机的操作，可以调用或提供根据本发明的方法和/或技术方案。而调用本发明的方法的程序指令，可能被存储在固定的或可移动的记录介质中，和/或通过广播或其他信号承载媒体中的数据流而被传输，和/或被存储在根据所述程序指令运行的计算机设备的工作存储器中。在此，根据本发明的一个实施例包括一个装置，该装置包括用于存储计算机程序指令的存储器和用于执行程序指令的处理器，其中，当该计算机程序指令被该处理器执行时，触发该装置运行基于前述根据本发明的多个实施例的方法和/或技术方案。

对于本领域技术人员而言，显然本发明不限于上述示范性实施例的细节，而且在不背离本发明的精神或基本特征的情况下，能够以其他的具体形式实现本发明。因此，无论从哪一点来看，均应将实施例看作是示范性的，而且是非限制性的，本发明的范围由所附权利要求而不是上述说明限定，因此旨在将落在权利要求的等同要件的含义和范围内的所有变化涵括在本发明内。不应将权利要求中的任何附图标记视为限制所涉及的权利要求。此外，显然“包括”一词不排除其他单元或步骤，单数不排除复数。装置权利要求中陈述的多个单元或装置也可以由一个单元或装置通过软件或者硬件来实现。第一，第二等词语用来表示名称，而并不表示任何特定的顺序。

权 利 要 求 书

1. 一种刷新磁盘输入输出请求的处理方法, 其中, 该方法包括:

在每一个磁盘输入输出请求到来时, 为每个磁盘输入输出请求递增地分配唯一的标识号, 其中, 所述磁盘输入输出请求包括磁盘写输入输出请求和刷新磁盘输入输出请求
5 中的任一种;

判断所述到来的磁盘输入输出请求的种类,

若为刷新磁盘输入输出请求, 则判断所述到来的刷新磁盘输入输出请求是否满足预设条件, 根据判断结果确定是否处理所述到来的刷新磁盘输入输出请求。

2. 如权利要求 1 所述的方法, 其中, 判断所述到来的刷新磁盘输入输出请求是否满足预设条件, 根据判断结果确定是否处理所述到来的刷新磁盘输入输出请求包括:
10

判断所述到来的刷新磁盘输入输出请求是否携带数据,

若携带数据, 将实时磁盘写输入输出请求的累计个数记录为原始累计个数后, 将实时磁盘写输入输出请求的累计个数更新为 1; 若没有携带数据, 将实时磁盘写输入输出请求的累计个数记录为原始累计个数后, 将实时磁盘写输入输出请求的累计个数更新为
15 0, 其中, 所述实时磁盘写输入输出请求的累计个数每次记录从上一个刷新磁盘输入输出请求至当前时刻所到达的磁盘写输入输出请求的累计个数;

根据所述原始累计个数确定是否处理所述到来的刷新磁盘输入输出请求。

3. 如权利要求 2 所述的方法, 其中, 根据所述原始累计个数确定是否处理所述到来的刷新磁盘输入输出请求, 包括:

20 判断所述原始累计个数是否为 0, 且此时键码键值关联容器是否为空, 其中, 所述键码键值关联容器用于存放刷新磁盘输入输出请求的标识号和对应的节点信息, 每个节点信息包括一个刷新磁盘输入输出请求和该刷新磁盘输入输出请求的累计间隙数, 所述累计间隙数的初始值为所述原始累计个数;

若所述原始累计个数为 0, 且此时键码键值关联容器为空, 立即处理该刷新磁盘输入输出请求;
25

若所述原始累计个数不为 0, 或此时键码键值关联容器不为空, 则将该到来的刷新磁盘输入输出请求的标识号和对应的节点信息插入所述键码键值关联容器中。

4. 如权利要求 3 所述的方法, 其中, 所述键码键值关联容器为红黑树容器。

5. 如权利要求 3 所述的方法, 其中, 判断所述到来的磁盘输入输出请求的种类之后,
30 还包括:

若为磁盘写输入输出请求，将所述到来的磁盘写输入输出请求的标识号与上一个刷新磁盘输入输出请求的标识号比较，

若该磁盘写输入输出请求的标识号大于上一个刷新磁盘输入输出请求的标识号，则将所述实时磁盘写输入输出请求的累计个数加 1；

5 若该磁盘写输入输出请求的标识号小于上一个刷新磁盘输入输出请求的标识号，则从所述键码键值关联容器中找到恰大于该磁盘写输入输出请求的标识号的刷新磁盘输入输出请求，并将该找到的刷新磁盘输入输出请求的累计间隙数加 1。

6. 如权利要求 3 至 5 所述的方法，其中，根据判断结果确定是否处理所述到来的刷新磁盘输入输出请求之后，还包括：

10 当一个磁盘输入输出请求的数据成功转发到实际存储介质中后，将该磁盘输入输出请求的标识号与上一个刷新磁盘输入输出请求的标识号进行比较，根据比较结果处理该磁盘输入输出请求，其中，磁盘输入输出请求包括磁盘写输入输出请求或带数据的刷新磁盘输入输出请求。

7. 如权利要求 6 所述的方法，其中，将该磁盘输入输出请求的标识号与上一个刷新磁盘输入输出请求的标识号进行比较，根据比较结果处理该磁盘输入输出请求，包括：

15 若该磁盘输入输出请求的标识号大于上一个刷新磁盘输入输出请求的标识号，则将实时磁盘写输入输出请求的累计个数减 1。

8. 如权利要求 7 所述的方法，其中，将该磁盘输入输出请求的标识号与上一个刷新磁盘输入输出请求的标识号进行比较，根据比较结果处理该磁盘输入输出请求，还包括：

20 若该磁盘输入输出请求的标识号小于等于上一个刷新磁盘输入输出请求的标识号，则在所述键码键值关联容器中查找恰大于该磁盘输入输出请求的标识号的刷新磁盘输入输出请求，并将该找到的刷新磁盘输入输出请求的累计间隙数减 1 后，判断该找到的刷新磁盘输入输出请求的当前的累计间隙数是否为 0，

25 如果为 0，判断该找到的刷新磁盘输入输出请求的标识号是否是所述键码键值关联容器中最小的标识号，

若是，处理该找到的刷新磁盘输入输出请求后，从键码键值关联容器中删除该找到的刷新磁盘输入输出请求的标识号和对应的节点信息；

重复如下步骤，直至查找到的刷新磁盘输入输出请求的累计间隙数大于 0；

30 查找所述键码键值关联容器中标识号紧随于最近删除的刷新磁盘输入输出请求的标识号之后的刷新磁盘输入输出请求，判断该查找到的刷新磁盘输入输出请求的累计间

隙数是否也为 0，若为 0，则处理该查找到的刷新磁盘输入输出请求后，从所述键码键值关联容器中删除该查找到的刷新磁盘输入输出请求的标识号和对应的节点信息。

9. 一种刷新磁盘输入输出请求的处理设备，其中，该设备包括：

分配标识号装置，用于在每一个磁盘输入输出请求到来时，为每个磁盘输入输出请求递增地分配唯一的标识号，其中，所述磁盘输入输出请求包括磁盘写输入输出请求和刷新磁盘输入输出请求中的任一种；

种类判断装置，用于判断所述到来的磁盘输入输出请求的种类，

条件判断装置，用于若为刷新磁盘输入输出请求，则判断所述到来的刷新磁盘输入输出请求是否满足预设条件，根据判断结果确定是否处理所述到来的刷新磁盘输入输出请求。

10. 如权利要求 9 所述的设备，其中，所述条件判断装置，用于判断所述到来的刷新磁盘输入输出请求是否携带数据，

若携带数据，将实时磁盘写输入输出请求的累计个数记录为原始累计个数后，将实时磁盘写输入输出请求的累计个数更新为 1；若没有携带数据，将实时磁盘写输入输出请求的累计个数记录为原始累计个数后，将实时磁盘写输入输出请求的累计个数更新为 0，其中，所述实时磁盘写输入输出请求的累计个数每次记录从上一个刷新磁盘输入输出请求至当前时刻所到达的磁盘写输入输出请求的累计个数；

根据所述原始累计个数确定是否处理所述到来的刷新磁盘输入输出请求。

11. 如权利要求 10 所述的设备，其中，所述条件判断装置，用于判断所述原始累计个数是否为 0，且此时键码键值关联容器是否为空，其中，所述键码键值关联容器用于存放刷新磁盘输入输出请求的标识号和对应的节点信息，每个节点信息包括一个刷新磁盘输入输出请求和该刷新磁盘输入输出请求的累计间隙数，所述累计间隙数的初始值为所述原始累计个数；

若所述原始累计个数为 0，且此时键码键值关联容器为空，立即处理该刷新磁盘输入输出请求；

若所述原始累计个数不为 0，或此时键码键值关联容器不为空，则将该到来的刷新磁盘输入输出请求的标识号和对应的节点信息插入所述键码键值关联容器中。

12. 如权利要求 11 所述的设备，其中，所述键码键值关联容器为红黑树容器。

13. 如权利要求 11 所述的设备，其中，所述设备还包括：

磁盘写输入输出请求处理装置，用于若为磁盘写输入输出请求，将所述到来的磁盘

写输入输出请求的标识号与上一个刷新磁盘输入输出请求的标识号比较，

若该磁盘写输入输出请求的标识号大于上一个刷新磁盘输入输出请求的标识号，则将所述实时磁盘写输入输出请求的累计个数加 1；

若该磁盘写输入输出请求的标识号小于上一个刷新磁盘输入输出请求的标识号，则
5 从所述键码键值关联容器中找到恰大于该磁盘写输入输出请求的标识号的刷新磁盘输入输出请求，并将该找到的刷新磁盘输入输出请求的累计间隙数加 1。

14. 如权利要求 11 至 13 所述的设备，其中，所述设备还包括：

标识号比较装置，用于当一个磁盘输入输出请求的数据成功转发到实际存储介质中后，将该磁盘输入输出请求的标识号与上一个刷新磁盘输入输出请求的标识号进行比较，根据比较结果处理该磁盘输入输出请求，其中，磁盘输入输出请求包括磁盘写输入
10 输出请求或带数据的刷新磁盘输入输出请求。

15. 如权利要求 14 所述的设备，其中，所述标识号比较装置，用于若该磁盘输入输出请求的标识号大于上一个刷新磁盘输入输出请求的标识号，则将实时磁盘写输入输出请求的累计个数减 1。

15 16. 如权利要求 15 所述的设备，其中，所述标识号比较装置，还用于若该磁盘输入输出请求的标识号小于等于上一个刷新磁盘输入输出请求的标识号，则在所述键码键值关联容器中查找恰大于该磁盘输入输出请求的标识号的刷新磁盘输入输出请求，并将该找到的刷新磁盘输入输出请求的累计间隙数减 1 后，判断该找到的刷新磁盘输入输出请求的当前的累计间隙数是否为 0，

20 如果为 0，判断该找到的刷新磁盘输入输出请求的标识号是否是所述键码键值关联容器中最小的标识号，

若是，处理该找到的刷新磁盘输入输出请求后，从键码键值关联容器中删除该找到的刷新磁盘输入输出请求的标识号和对应的节点信息；

重复查找所述键码键值关联容器中标识号紧随于最近删除的刷新磁盘输入输出请求的标识号之后的刷新磁盘输入输出请求，判断该查找到的刷新磁盘输入输出请求的累计间隙数是否也为 0，若为 0，则处理该查找到的刷新磁盘输入输出请求后，从所述键码键值关联容器中删除该查找到的刷新磁盘输入输出请求的标识号和对应的节点信息，直至查找到的刷新磁盘输入输出请求的累计间隙数大于 0。

25

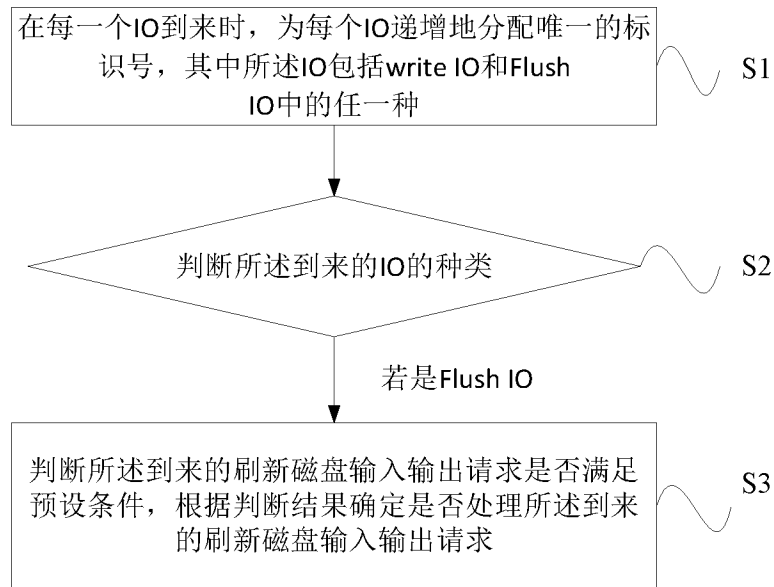


图 1

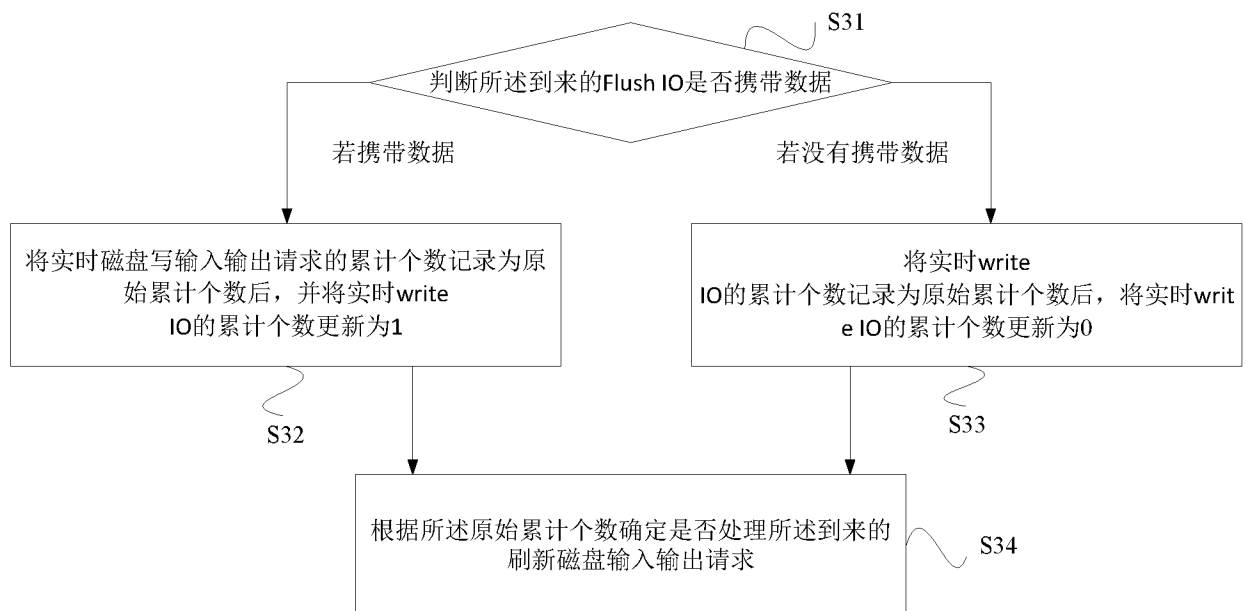


图 2

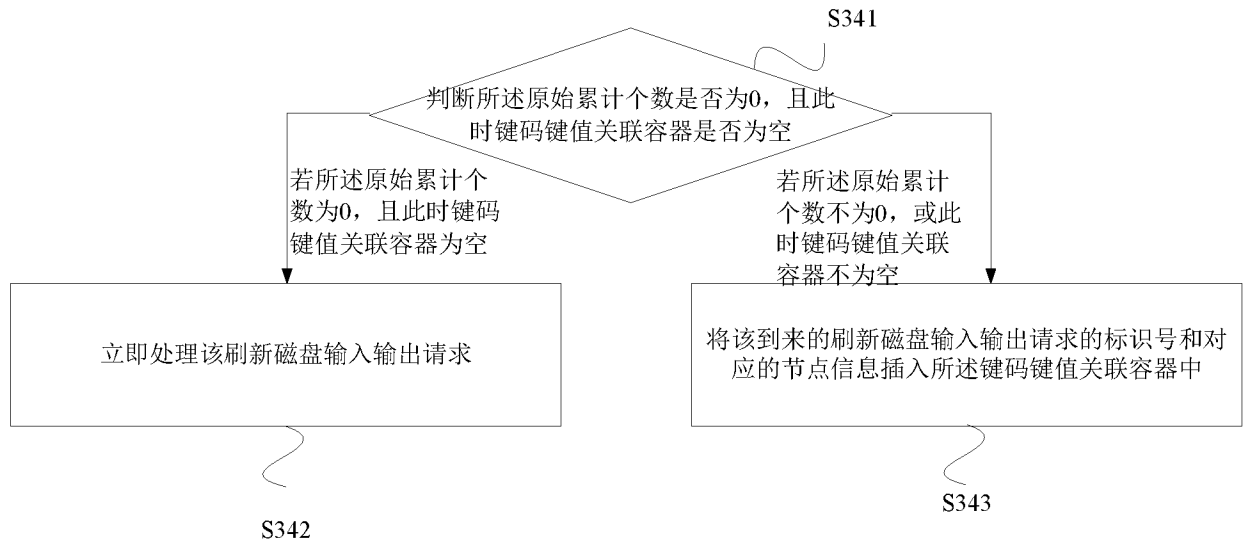


图 3

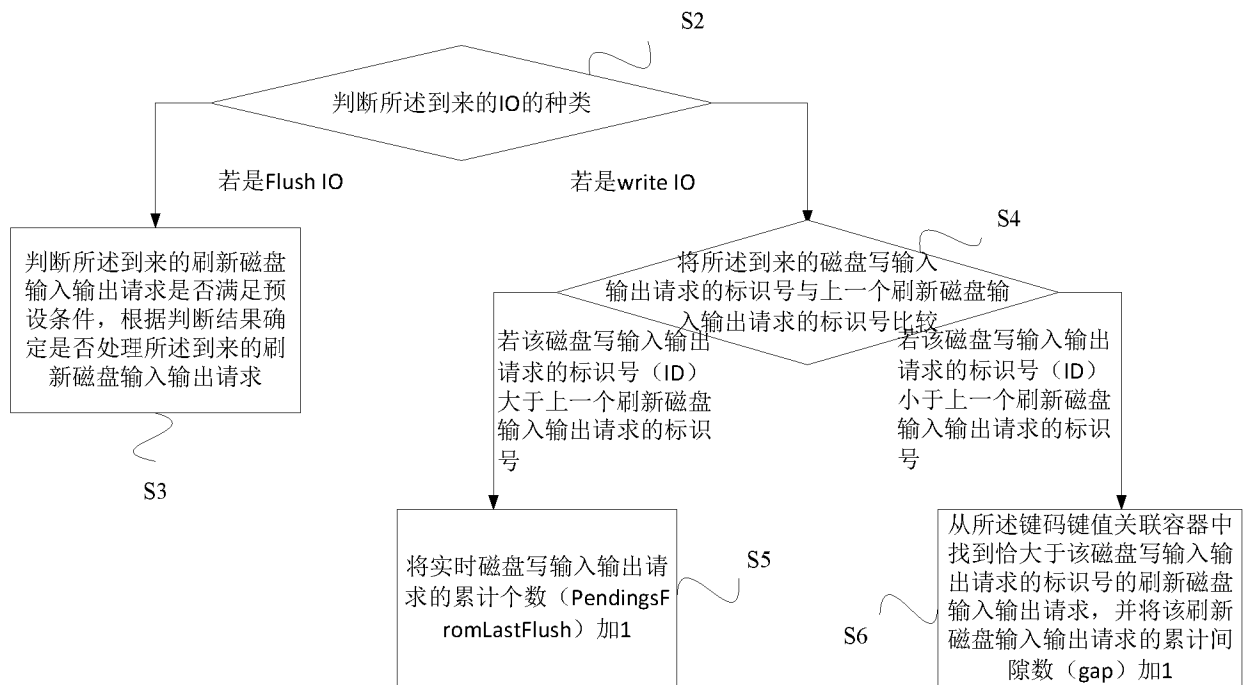


图 4

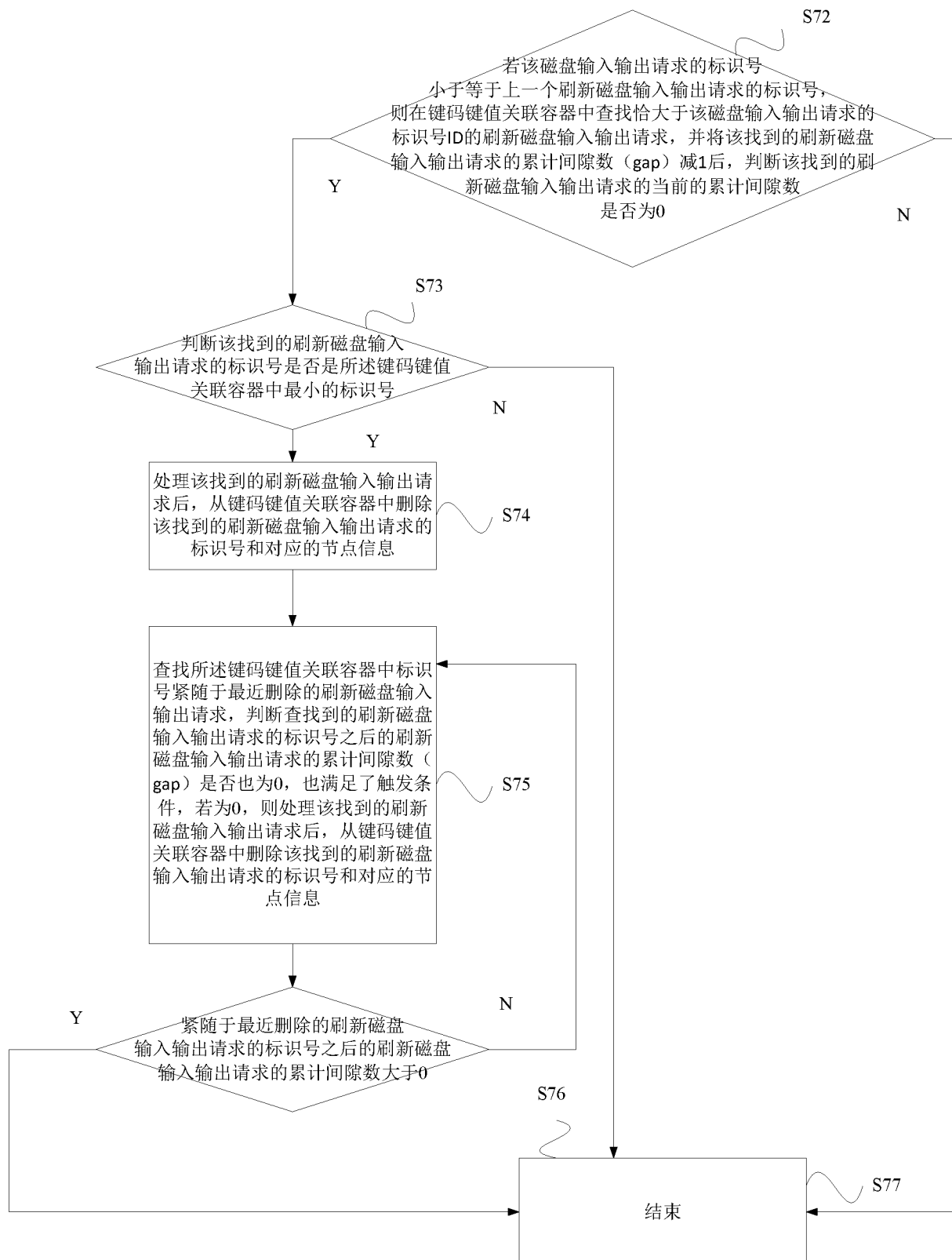


图 5

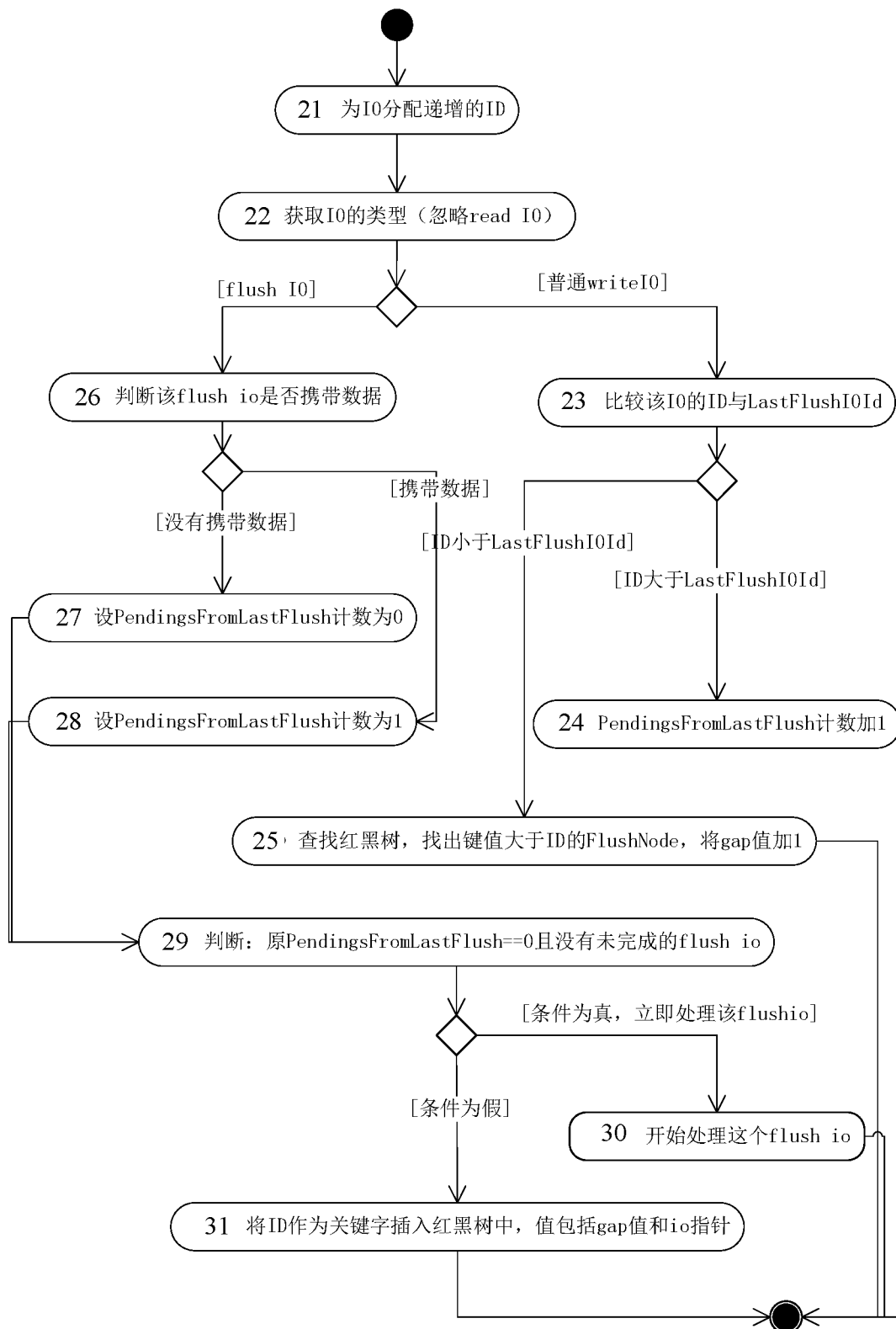


图 6

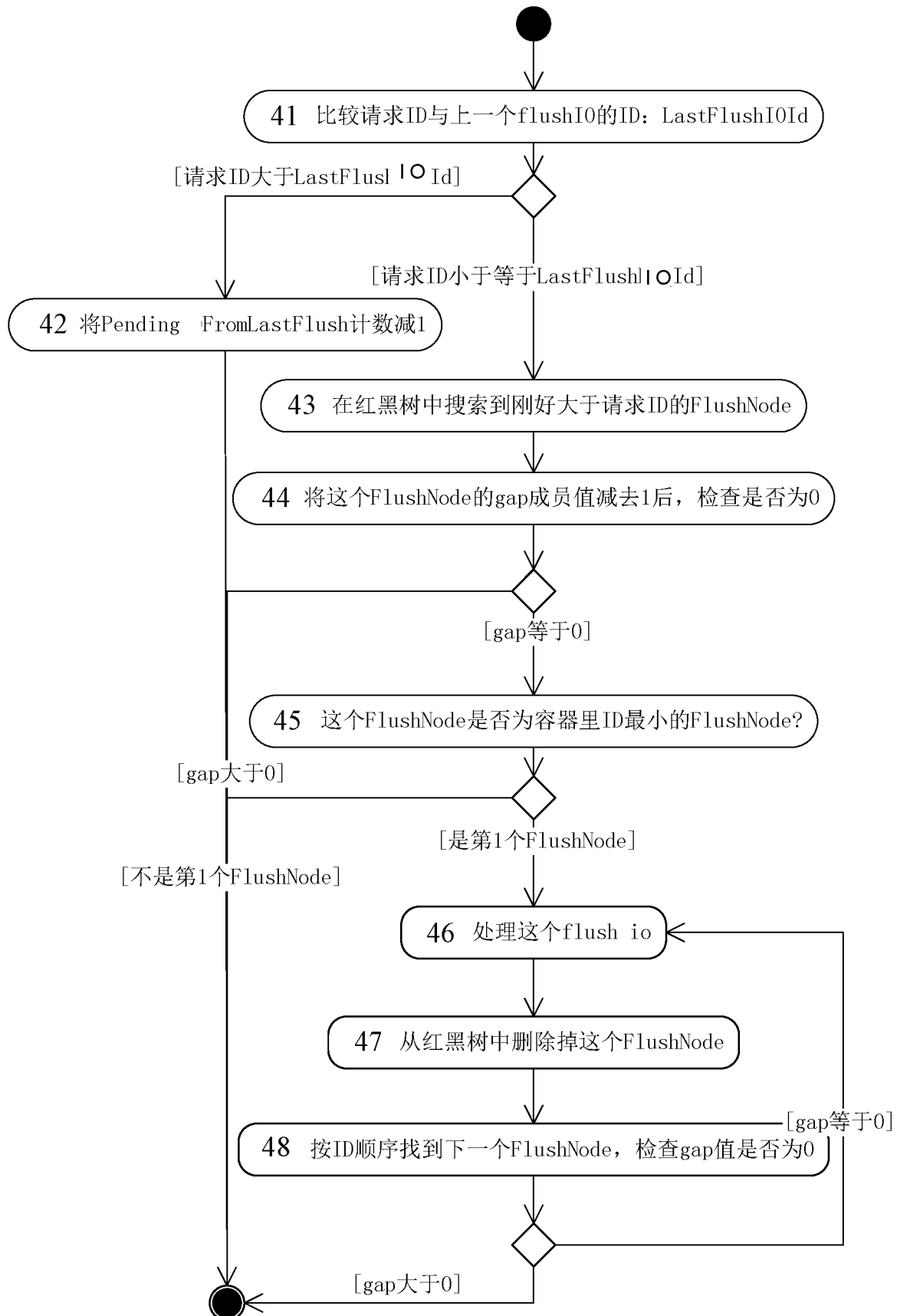


图 7

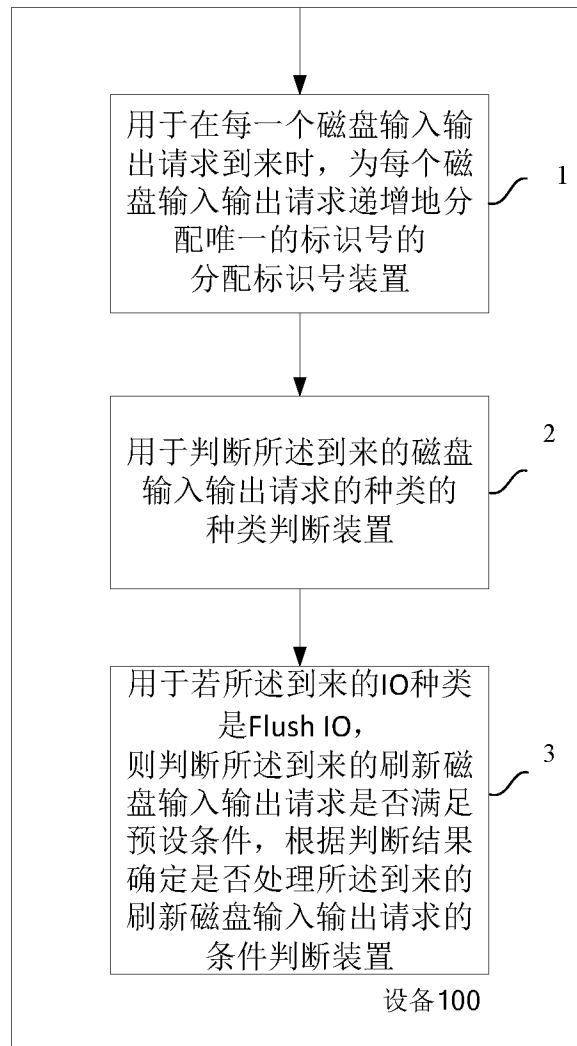


图 8

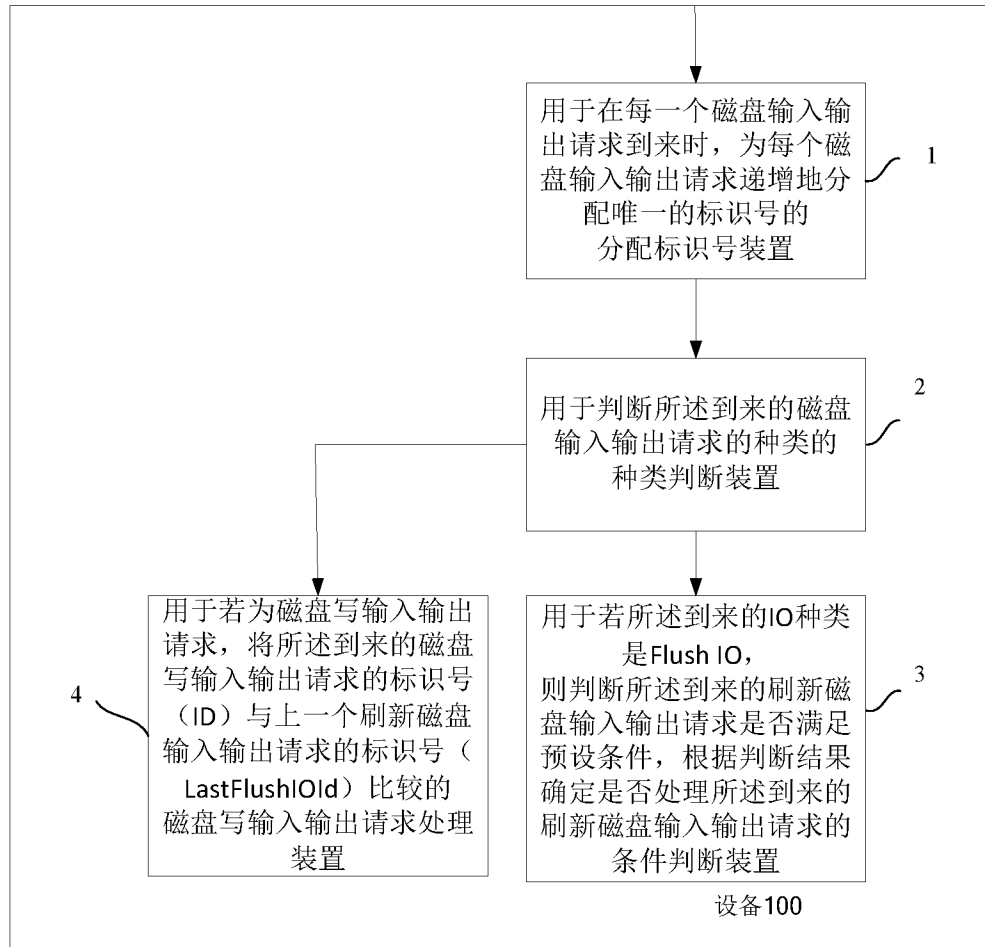


图 9

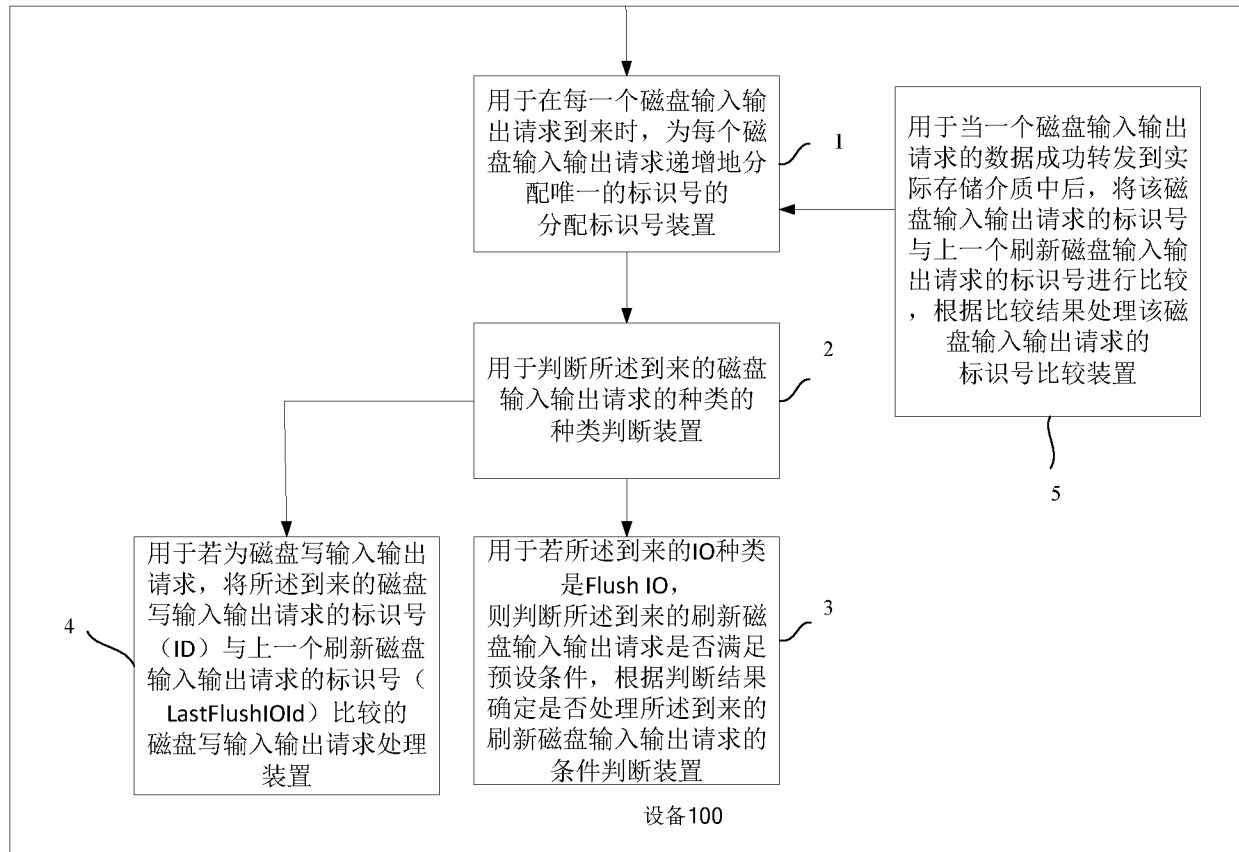


图 10

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2016/098008

A. CLASSIFICATION OF SUBJECT MATTER

G06F 3/06 (2006.01) i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

CNPAT, WPI, EPODOC, CNKI, GOOGLE: hard disk, read-write, refresh, virtual machine, disk?, IO, input, output, request?, flush, virtual, machine, VM

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	LI, Dingding, "Write Performance Optimization of Virtual Machine on The Local Storage", ELECTRONIC TECHNOLOGY & INFORMATION SCIENCE, CHINA DOCTOR DISSERTATION FULL-TEXT DATABASE (ELECTRONIC JOURNAL), no. 02, 15 February 2014 (15.02.2014), page 24, paragraph 4 to page 29, paragraph 1	1, 9
A	CN 103823636 A (EJITEC CO., LTD.), 28 May 2014 (28.05.2014), description, paragraphs 10-16 and 65-68	1-16
A	CN 102541982 A (DAWNING INFORMATION INDUSTRY (BEIJING) CO., LTD.), 04 July 2012 (04.07.2012), the whole document	1-16
A	CN 104407820 A (HUAWEI TECHNOLOGIES CO., LTD.), 11 March 2015 (11.03.2015), the whole document	1-16
A	CN 103049219 A (HUAZHONG UNIVERSITY OF SCIENCE AND TECHNOLOGY), 17 April 2013 (17.04.2013), the whole document	1-16
A	CN 103336669 A (HUAZHONG UNIVERSITY OF SCIENCE AND TECHNOLOGY), 02 October 2013 (02.10.2013), the whole document	1-16

☒ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents:	"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A" document defining the general state of the art which is not considered to be of particular relevance	"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E" earlier application or patent but published on or after the international filing date	"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&" document member of the same patent family
"O" document referring to an oral disclosure, use, exhibition or other means	
"P" document published prior to the international filing date but later than the priority date claimed	

Date of the actual completion of the international search 19 October 2016 (19.10.2016)	Date of mailing of the international search report 29 November 2016 (29.11.2016)
Name and mailing address of the ISA/CN: State Intellectual Property Office of the P. R. China No. 6, Xitucheng Road, Jimenqiao Haidian District, Beijing 100088, China Facsimile No.: (86-10) 62019451	Authorized officer BIAN, Zhen Telephone No.: (86-10) 61648106

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2016/098008**C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT**

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	CN 104375883 A (INSPUR ELECTRONIC INFORMATION INDUSTRY CO., LTD.), 25 February 2015 (25.02.2015), the whole document	1-16
A	US 2009119538 A1 (VMWAR, INC.), 07 May 2009 (07.05.2009), the whole document	1-16
A	US 2006069716 A1 (INTERNATIONAL BUSINESS MACHINES CORPORATION), 30 March 2006 (30.03.2006), the whole document	1-16

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/CN2016/098008

Patent Documents referred in the Report	Publication Date	Patent Family	Publication Date
CN 103823636 A	28 May 2014	None	
CN 102541982 A	04 July 2012	None	
CN 104407820 A	11 March 2015	None	
CN 103049219 A	17 April 2013	None	
CN 103336669 A	02 October 2013	None	
CN 104375883 A	25 February 2015	None	
US 2009119538 A1	07 May 2009	US 2009119665 A1	07 May 2009
US 2006069716 A1	30 March 2006	None	

国际检索报告

国际申请号

PCT/CN2016/098008

A. 主题的分类

G06F 3/06(2006.01)i

按照国际专利分类(IPC)或者同时按照国家分类和IPC两种分类

B. 检索领域

检索的最低限度文献(标明分类系统和分类号)

G06F

包含在检索领域中的除最低限度文献以外的检索文献

在国际检索时查阅的电子数据库(数据库的名称, 和使用的检索词(如使用))

CNPAT, WPI, EPODOC, CNKI, GOOGLE: 磁盘, 硬盘, 输入, 输出, 读写, 请求, 刷新, 虚拟机, disk?, IO, input, output, request?, flush, virtual, machine, VM

C. 相关文件

类 型*	引用文件, 必要时, 指明相关段落	相关的权利要求
X	李丁丁. “虚拟机本地存储写性能优化研究” 中国博士学位论文全文数据库(电子期刊) 信息科技辑, 第02期, 2014年 2月 15日(2014 - 02 - 15), 第24页第4段到第29页第1段	1、9
A	CN 103823636 A (苏州捷泰科信息技术有限公司) 2014年 5月 28日(2014 - 05 - 28) 说明书第10-16、65-68段	1-16
A	CN 102541982 A (曙光信息产业北京有限公司) 2012年 7月 4日(2012 - 07 - 04) 全文	1-16
A	CN 104407820 A (华为技术有限公司) 2015年 3月 11日(2015 - 03 - 11) 全文	1-16
A	CN 103049219 A (华中科技大学) 2013年 4月 17日(2013 - 04 - 17) 全文	1-16
A	CN 103336669 A (华中科技大学) 2013年 10月 2日(2013 - 10 - 02) 全文	1-16

☒ 其余文件在C栏的续页中列出。☒ 见同族专利附件。

* 引用文件的具体类型:

“A” 认为不特别相关的表示了现有技术一般状态的文件

“E” 在国际申请日的当天或之后公布的在先申请或专利

“L” 可能对优先权要求构成怀疑的文件, 或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件(如具体说明的)

“O” 涉及口头公开、使用、展览或其他方式公开的文件

“P” 公布日先于国际申请日但迟于所要求的优先权日的文件

“T” 在申请日或优先权日之后公布, 与申请不相抵触, 但为了理解发明之理论或原理的在后文件

“X” 特别相关的文件, 单独考虑该文件, 认定要求保护的发明不是新颖的或不具有创造性

“Y” 特别相关的文件, 当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时, 要求保护的发明不具有创造性

“&” 同族专利的文件

国际检索实际完成的日期

2016年 10月 19日

国际检索报告邮寄日期

2016年 11月 29日

ISA/CN的名称和邮寄地址

中华人民共和国国家知识产权局(ISA/CN)
中国北京市海淀区蓟门桥西土城路6号 100088

受权官员

边臻

传真号 (86-10)62019451

电话号码 (86-10)61648106

C. 相关文件

类 型*	引用文件，必要时，指明相关段落	相关的权利要求
A	CN 104375883 A (浪潮电子信息产业股份有限公司) 2015年 2月 25日 (2015 - 02 - 25) 全文	1-16
A	US 2009119538 A1 (VMWAR, INC.) 2009年 5月 7日 (2009 - 05 - 07) 全文	1-16
A	US 2006069716 A1 (INTERNATIONAL BUSINESS MACHINES CORPORATION) 2006年 3月 30日 (2006 - 03 - 30) 全文	1-16

国际检索报告
关于同族专利的信息

国际申请号
PCT/CN2016/098008

检索报告引用的专利文件			公布日 (年/月/日)	同族专利	公布日 (年/月/日)
CN	103823636	A	2014年 5月 28日	无	
CN	102541982	A	2012年 7月 4日	无	
CN	104407820	A	2015年 3月 11日	无	
CN	103049219	A	2013年 4月 17日	无	
CN	103336669	A	2013年 10月 2日	无	
CN	104375883	A	2015年 2月 25日	无	
US	2009119538	A1	2009年 5月 7日	US 2009119665 A1	2009年 5月 7日
US	2006069716	A1	2006年 3月 30日	无	

表 PCT/ISA/210 (同族专利附件) (2009年7月)