



(12)发明专利申请

(10)申请公布号 CN 111104066 A

(43)申请公布日 2020.05.05

(21)申请号 201911304631.1

(22)申请日 2019.12.17

(71)申请人 华中科技大学

地址 430070 湖北省武汉市洪山区珞喻路
1037号

申请人 腾讯科技(深圳)有限公司

(72)发明人 张煜 周可 王桦 胡健鹰
吉永光(74)专利代理机构 深圳市深佳知识产权代理事
务所(普通合伙) 44285

代理人 王兆林

(51)Int.Cl.

G06F 3/06(2006.01)

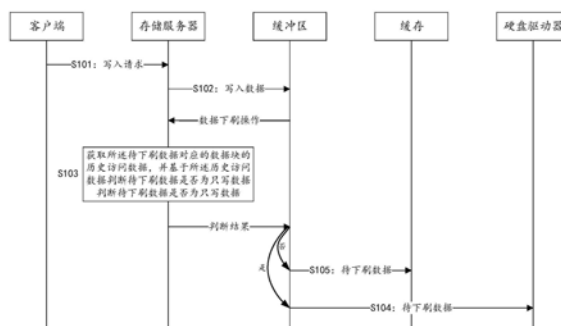
权利要求书3页 说明书13页 附图7页

(54)发明名称

数据写入方法、装置及存储服务器和计算机
可读存储介质

(57)摘要

本申请公开了一种数据写入方法、装置及一种存储服务器和计算机可读存储介质,该方法包括:当接收到写入请求时,将所述写入请求对应的写入数据写入写缓冲区中;当触发所述写缓冲区的数据下刷操作时,获取所述待下刷数据对应的数据块的历史访问数据;基于所述历史访问数据判断待下刷数据是否为只写数据;若是,则将所述待下刷数据写入硬盘驱动器中;若否,则将所述待下刷数据写入缓存中。本申请提供的数据写入方法,可以有效减少写入缓存的无用流量,同时在缓存中为普通数据留出更多空间,提高缓存空间利用率,从而提高缓存的读取命中率,提升存储系统的读性能,实现了对缓存的最低写入流量,提高了写策略的效率。



1. 一种数据写入方法,其特征在于,包括:
当接收到写入请求时,将所述写入请求对应的写入数据写入写缓冲区中;
当触发所述写缓冲区的数据下刷操作时,获取所述待下刷数据对应的数据块的历史访问数据;
基于所述历史访问数据判断待下刷数据是否为只写数据;
若是,则将所述待下刷数据写入硬盘驱动器中;
若否,则将所述待下刷数据写入缓存中。
2. 根据权利要求1所述数据写入方法,其特征在于,将所述待下刷数据写入硬盘驱动器中,包括:
将所述缓存中所述待下刷数据对应的脏数据刷新至所述硬盘驱动器;
将所述待下刷数据写入硬盘驱动器中。
3. 根据权利要求1所述数据写入方法,其特征在于,将所述待下刷数据写入缓存中,包括:
将所述待下刷数据写入缓存中;
当满足预设条件时,将所述缓存中的数据刷新至所述硬盘驱动器中。
4. 根据权利要求1所述数据写入方法,其特征在于,将所述待下刷数据写入缓存中,包括:
将所述待下刷数据同时写入所述缓存和所述硬盘驱动器中。
5. 根据权利要求1所述数据写入方法,其特征在于,所述当触发写缓冲区的数据下刷操作时,获取所述待下刷数据对应的数据块的历史访问数据,包括:
当所述写缓冲区的数据量达到阈值时,触发写缓冲区的数据下刷操作,并获取所述待下刷数据对应的数据块的历史访问数据。
6. 根据权利要求1至5中任一项所述数据写入方法,其特征在于,所述基于所述历史访问数据判断待下刷数据是否为只写数据,包括:
在所述历史访问数据中提取所述待下刷数据对应的数据块的特征信息作为第一特征信息,将所述第一特征信息输入训练完成的分类器中以便判断待下刷数据是否为只写数据。
7. 根据权利要求6所述数据写入方法,其特征在于,还包括:
当所述缓存触发数据替换操作时,基于替换数据的历史读取数据对所述替换数据进行标记;
获取所述替换数据对应的数据块的特征信息作为第二特征信息,并利用标记完成的替换数据和所述第二特征信息训练分类器得到所述训练完成的分类器。
8. 根据权利要求7所述数据写入方法,其特征在于,所述获取所述替换数据对应的数据块的特征信息作为第二特征信息,包括:
确定所述替换数据对应的数据块所属的数据区域;其中,所述数据区域包括多个数据块,所述数据区域的大小为访问粒度的整数倍;
提取所述数据区域的特征信息作为所述第二特征信息。
9. 根据权利要求6所述数据写入方法,其特征在于,所述特征信息包括时间特征和/或请求类特征;所述时间特征为描述访问数据块的时间的特征,所述请求类特征为描述访问

数据块的请求的特征。

10. 根据权利要求9所述数据写入方法,其特征在于,所述时间特征包括上次访问时间戳和/或平均重用时间差;所述上次访问时间戳描述上一次访问数据块的时间点,所述平均重用时间差描述访问数据块的平均时间间隔;

所述请求类特征包括平均请求大小、大请求比率、小请求比率和写请求比率中的任一项或任几项的组合;所述大请求比率描述访问数据块的请求中请求大小大于第一预设值的请求的比率,所述小请求比率描述访问数据块的请求中请求大小小于第二预设值的请求的比率,所述写请求比率描述访问数据块的请求中写请求的比率,所述第一预设值大于或等于所述第二预设值。

11. 根据权利要求1至5中任一项所述数据写入方法,其特征在于,所述基于所述历史访问数据判断待下刷数据是否为只写数据,包括:

基于所述历史访问数据利用统计学方法判断待下刷数据是否为只写数据。

12. 根据权利要求11所述数据写入方法,其特征在于,所述基于所述历史访问数据利用统计学方法判断待下刷数据是否为只写数据,包括:

基于所述历史访问数据确定多个时间窗口内所述待下刷数据对应的数据块的写请求次数,并根据所有所述写请求次数计算所述待下刷数据对应的数据块的只写概率;其中,所述只写概率描述所述数据块存储只写数据的概率;

判断所述只写概率是否大于第三预设值,若是,则判定所述待下刷数据为所述只写数据,若否,则判定所述待下刷数据为所述普通数据。

13. 一种数据写入装置,其特征在于,包括:

第一写入模块,用于当接收到写入请求时,将所述写入请求对应的写入数据写入写缓冲区中;

获取模块,用于当触发所述写缓冲区的数据下刷操作时,获取所述待下刷数据对应的数据块的历史访问数据;

判断模块,用于基于所述历史访问数据判断待下刷数据是否为只写数据;

第二写入模块,用于当所述待下刷数据为只写数据时,将所述待下刷数据写入硬盘驱动器中;

第三写入模块,用于当所述待下刷数据为非只写数据时,将所述待下刷数据写入缓存中。

14. 一种存储服务器,其特征在于,包括:

处理器和存储器;

其中,所述处理器用于执行所述存储器中存储的程序;

所述存储器用于存储程序,所述程序至少用于:

当接收到写入请求时,将所述写入请求对应的写入数据写入写缓冲区中;

当触发所述写缓冲区的数据下刷操作时,获取所述待下刷数据对应的数据块的历史访问数据;

基于所述历史访问数据判断待下刷数据是否为只写数据;

若是,则将所述待下刷数据写入硬盘驱动器中;

若否,则将所述待下刷数据写入缓存中。

15. 一种计算机可读存储介质,其特征在于,所述计算机可读存储介质上存储有计算机程序,所述计算机程序被处理器执行时实现如权利要求1至12任一项所述数据写入方法的步骤。

数据写入方法、装置及存储服务器和计算机可读存储介质

技术领域

[0001] 本申请涉及存储技术领域,更具体地说,涉及数据写入方法、装置及存储服务器和计算机可读存储介质。

背景技术

[0002] 当客户端产生写请求时,存储服务器依据写策略决定是否将待写入数据加载到缓存以及何时将缓存中的脏数据下刷到后端的硬盘驱动器。因此,写策略对系统性能、数据一致性以及对高速缓存介质的写入流量具有很大影响。

[0003] 在相关技术中,存在三个基本写策略:写穿策略(write-through)、写回策略(write-back)和绕写策略(write-around)。写穿策略为将待写入数据同时写入缓存和后端的硬盘驱动器,写回策略为将待写入数据首先写入缓存,然后异步下刷到后端的硬盘驱动器,绕写策略为将待写入数据直接写后端的硬盘驱动器,只有当存在读操作时才将未命中数据加载到缓存。

[0004] 在不断变化的多样化云工作负载下,固定的写策略注定是低效的。因此,如何提高写策略的效率是本领域技术人员需要解决的技术问题。

发明内容

[0005] 本申请的目的在于提供一种数据写入方法、装置及一种存储服务器和一种计算机可读存储介质,提高写策略的效率。

[0006] 为实现上述目的,本申请第一方面提供了一种数据写入方法,包括:

[0007] 当接收到写入请求时,将所述写入请求对应的写入数据写入写缓冲区中;

[0008] 当触发所述写缓冲区的数据下刷操作时,获取所述待下刷数据对应的数据块的历史访问数据;

[0009] 基于所述历史访问数据判断待下刷数据是否为只写数据;

[0010] 若是,则将所述待下刷数据写入硬盘驱动器中;

[0011] 若否,则将所述待下刷数据写入缓存中。

[0012] 结合本申请的第一方面,在本申请第一方面的第一种实施方式中,所述基于所述历史访问数据判断待下刷数据是否为只写数据,包括:

[0013] 在所述历史访问数据中提取所述待下刷数据对应的数据块的特征信息作为第一特征信息,将所述第一特征信息输入训练完成的分类器中以便判断待下刷数据是否为只写数据。

[0014] 结合本申请的第一方面的第一种实施方式,在本申请第一方面的第二种实施方式中,还包括:

[0015] 当所述缓存触发数据替换操作时,基于替换数据的历史读取数据对所述替换数据进行标记;

[0016] 获取所述替换数据对应的数据块的特征信息作为第二特征信息,并利用标记完成

的替换数据和所述第二特征信息训练分类器得到所述训练完成的分类器。

[0017] 结合本申请的第一方面的第二种实施方式,在本申请第一方面的第三种实施方式中,所述获取所述替换数据对应的数据块的特征信息作为第二特征信息,包括:

[0018] 确定所述替换数据对应的数据块所属的数据区域;其中,所述数据区域包括多个数据块,所述数据区域的大小为访问粒度的整数倍;

[0019] 提取所述数据区域的特征信息作为所述第二特征信息。

[0020] 结合本申请的第一方面的第一种实施方式,在本申请第一方面的第四种实施方式中,所述特征信息包括时间特征和/或请求类特征;所述时间特征为描述访问数据块的时间的特征,所述请求类特征为描述访问数据块的请求的特征。

[0021] 为实现上述目的,本申请第二方面提供了一种数据写入装置,包括:

[0022] 第一写入模块,用于当接收到写入请求时,将所述写入请求对应的写入数据写入写缓冲区中;

[0023] 获取模块,用于当触发所述写缓冲区的数据下刷操作时,获取所述待下刷数据对应的数据块的历史访问数据;

[0024] 判断模块,用于基于所述历史访问数据判断待下刷数据是否为只写数据;

[0025] 第二写入模块,用于当所述待下刷数据为只写数据时,将所述待下刷数据写入硬盘驱动器中;

[0026] 第三写入模块,用于当所述待下刷数据为非只写数据时,将所述待下刷数据写入缓存中。

[0027] 为实现上述目的,本申请第三方面提供了一种存储服务器,包括:

[0028] 处理器和存储器;

[0029] 其中,所述处理器用于执行所述存储器中存储的程序;

[0030] 所述存储器用于存储程序,所述程序至少用于:

[0031] 当接收到写入请求时,将所述写入请求对应的写入数据写入写缓冲区中;

[0032] 当触发所述写缓冲区的数据下刷操作时,获取所述待下刷数据对应的数据块的历史访问数据;

[0033] 基于所述历史访问数据判断待下刷数据是否为只写数据;

[0034] 若是,则将所述待下刷数据写入硬盘驱动器中;

[0035] 若否,则将所述待下刷数据写入缓存中。

[0036] 为实现上述目的,本申请第四方面提供了一种计算机可读存储介质,所述计算机可读存储介质上存储有计算机程序,所述计算机程序被处理器执行时实现如上述数据写入方法的步骤。

[0037] 通过以上方案可知,本申请提供了一种数据写入方法,包括:当接收到写入请求时,将所述写入请求对应的写入数据写入写缓冲区中;当触发所述写缓冲区的数据下刷操作时,获取所述待下刷数据对应的数据块的历史访问数据;基于所述历史访问数据判断待下刷数据是否为只写数据;若是,则将所述待下刷数据写入硬盘驱动器中;若否,则将所述待下刷数据写入缓存中。

[0038] 在本申请中,将客户端产生的数据划分为不同的类型,即只写数据和普通数据。只写数据在一定时间窗口内只存在写操作,而不存在读操作,也就是说,客户端的读请求在该

时间窗口内不会命中只写数据,将只写数据加载到缓存,不但不会提高缓存的读取命中率,还会对缓存造成大量非必要的写入流量。因此,本申请提供的数据写入方法,对只写数据和普通数据采取不同的写入方式,对于普通数据采用写穿策略或写回策略,将其写入缓存。而对于只写数据采用绕写策略,即将其直接写入后端的硬盘驱动器,可以有效减少写入缓存的无用流量,同时在缓存中为普通数据留出更多空间,提高缓存空间利用率,从而提高缓存的读取命中率,提升存储系统的读性能。由此可见,本申请提供的数据写入方法,实现了对缓存的最低写入流量,提高了写策略的效率。本申请还公开了一种数据写入装置及一种存储服务器和一种计算机可读存储介质,同样能实现上述技术效果。

[0039] 应当理解的是,以上的一般描述和后文的细节描述仅是示例性的,并不能限制本申请。

附图说明

[0040] 为了更清楚地说明本申请实施例或现有技术中的技术方案,下面将对实施例或现有技术描述中所需要使用的附图作简单的介绍,显而易见地,下面描述中的附图仅仅是本申请的一些实施例,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。附图是用来提供对本公开的进一步理解,并且构成说明书的一部分,与下面的具体实施方式一起用于解释本公开,但并不构成对本公开的限制。在附图中:

[0041] 图1为本申请实施例提供的一种数据写入系统的架构图;

[0042] 图2为本申请实施例提供的第一种数据写入方法的流程图;

[0043] 图3为本申请实施例提供的第二种数据写入方法的流程图;

[0044] 图4为本申请实施例提供的第三种数据写入方法的流程图;

[0045] 图5为本申请实施例提供的一种分类器训练方法的流程图;

[0046] 图6为本申请实施例提供的第四种数据写入方法的流程图;

[0047] 图7为本申请提供的一种应用实施例的系统架构图;

[0048] 图8为本申请提供的一种应用实施例的流程图;

[0049] 图9为本申请实施例提供的一种数据写入装置的结构图;

[0050] 图10为本申请实施例提供的一种存储服务器的结构图。

具体实施方式

[0051] 在典型的云块存储产品中,例如腾讯CBS、Ceph(分布式文件系统)等,客户端产生写入请求时,该写入请求对应的写入数据首先被写入写缓冲区,同时顺序记录在日志文件中。当写缓冲区中的脏数据量超过一定阈值时被提交到缓存,例如SSD(中文全称:固态驱动器,英文全称:Solid State Drive)缓存等,缓存中的写入数据采用写回策略异步下刷到后端的硬盘驱动器(英文全称:Hard Disk Drive,英文简称:HDD)存储。

[0052] 但是,本申请的发明人研究发现,CBS采集的一个月的IO(中文全称:输入/输出,英文全称:Input/Output)日志中,47.09%的数据只存在写操作,即47.09%的数据为只写数据。将只写数据提交到缓存而带来的加速效果用户并不能感知到,此时对于缓存,无论使用写回还是写穿策略,都会对缓存造成大量非必要的写入流量。另外,简单地部署绕写策略将

不可避免地降低读请求的命中率,严重降低服务质量。

[0053] 因此,在本申请中,将客户端产生的数据划分为只写数据和普通数据,只写数据在一定时间窗口内只存在写操作,采用绕写策略,对于普通数据采用写穿策略或写回策略,将其写入缓存。将只写数据直接写入后端的硬盘驱动器,可以有效减少写入缓存的无用流量,同时在缓存中为普通数据留出更多空间,提高缓存空间利用率。

[0054] 下面将结合本申请实施例中的附图,对本申请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申请一部分实施例,而不是全部的实施例。基于本申请中的实施例,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,都属于本申请保护的范围。

[0055] 为了便于理解本申请提供的数据写入的方法,首先对其使用的系统进行介绍。参见图1,其示出了本申请实施例提供的一种数据写入系统的架构图,如图1所示,包括客户端100、存储服务器200和存储区300,客户端100与存储服务器200、存储服务器200和存储区300之间通过网络400传输写入数据。

[0056] 存储区300包括写缓冲区31、缓存32和后端的硬盘驱动器(HDD)33,当客户端产生写入请求时,该写入请求对应的写入数据首先被写入写缓冲区31中,写缓冲区31可以由DRAM(中文全称:动态随机存取存储器,英文全称:Dynamic Random Access Memory)组成,在此不进行具体限定。写缓冲区31被触发数据下刷操作时,存储服务器200确定待下刷数据的数据类型,若其为只写数据,则将其直接写入后端的硬盘驱动器33,否则可以采用写穿策略将其同时写入缓存32和硬盘驱动器33,也可以采用写回策略为将其首先写入缓存32,然后异步下刷到硬盘驱动器33中,在此不进行具体限定。缓存32用于暂存数据,当客户端产生读请求时,若该读请求对应的数据在缓存32中时可以直接返回,不需要访问后端的硬盘驱动器33,提高读取效率。本实施例中的缓存32可以采用SSD缓存等,在此不进行具体限定。

[0057] 本申请实施例公开了一种数据写入方法,提高了写策略的效率。

[0058] 参见图2,本申请实施例提供的一种数据写入方法的流程图,如图2所示,包括:

[0059] S101:客户端产生写入请求,并将所述写入请求发送至存储服务器;

[0060] 在本步骤中,客户端产生写入请求,该写入请求包括写入数据,也可以在该写入请求中指定写入地址,在此不进行具体限定。客户端将写入请求发送至存储服务器,以便存储服务器依据预设定义的写策略对该写入请求进行处理。

[0061] S102:存储服务器将所述写入请求对应的写入数据写入写缓冲区中;

[0062] 在本步骤中,客户端产生的所有写入请求对应的写入数据首先被写入写缓冲区,同时顺序记录到日志文件中,日志文件用于数据恢复。

[0063] S103:当所述写缓冲区触发数据下刷操作时,存储服务器获取所述待下刷数据对应的数据块的历史访问数据,并基于所述历史访问数据判断待下刷数据是否为只写数据判断待下刷数据是否为只写数据;若是,则进入S104;若否,则进入S105;

[0064] 本步骤旨在确定待下刷数据的类型,只写数据或普通数据,以便后续步骤对不同类型的数据采用不同的写策略。只写数据为在一定时间窗口内只存在写操作而不存在读操作的数据,除只写数据之外的数据称之为普通数据。当写缓冲区触发数据下刷操作时,存储服务器确定待下刷数据,并获取该待下刷数据对应的数据块的历史访问数据。客户端在产生写入请求时,可以指定写入地址,此处的数据块即为该写入地址对应的数据块。历史访问

数据描述该数据块的历史访问情况,依据历史访问数据可以确定该数据块存储只写数据的概率,进而确定即将存储在该数据块内的待下刷数据是否为只写数据。需要说明的是,本实施例不对数据类型的具体确定方式进行限定,例如可以基于统计学方法或分类器等,均在本实施例的保护范围内。

[0065] 可以理解的是,本步骤不对数据下刷操作的触发条件进行限定,例如可以当写缓冲区的数据量达到阈值时,触发数据下刷操作,即所述当触发写缓冲区的数据下刷操作时,获取所述待下刷数据对应的数据块的历史访问数据的步骤包括:当所述写缓冲区的数据量达到阈值时,触发写缓冲区的数据下刷操作,并获取所述待下刷数据对应的数据块的历史访问数据。

[0066] S104:存储服务器将所述待下刷数据写入硬盘驱动器中;

[0067] 可以理解的是,缓存用于暂存数据,当客户端产生读请求时,若该读请求对应的数据在缓存中时可以直接返回,不需要访问后端的硬盘驱动器,提高读取效率。而只写数据在一定时间窗口内只存在写操作,而不存在读操作,可见,将只写数据加载到缓存,不但不会提高缓存的读取命中率,还会对缓存造成大量非必要的写入流量。

[0068] 因此,在本步骤中,对只写数据采用绕写策略,即当待下刷数据为只写数据时,将其直接写入后端的硬盘驱动器中。在缓存中为普通数据留出更多空间,提高缓存空间利用率,从而提高缓存的读取命中率。在具体实施中,为了进一步节省缓存空间,防止缓存中存在该待下刷数据对应的脏数据,保证数据一致性,首先查询缓存中是否存在该待下刷数据对应的脏数据,若是,则首先将缓存中脏数据刷新至硬盘驱动器,再将写缓冲区中的待下刷数据写入硬盘驱动器。即本步骤包括:将所述固态硬盘驱动器中所述待下刷数据对应的脏数据刷新至所述硬盘驱动器;将所述待下刷数据写入硬盘驱动器中。

[0069] S105:存储服务器将所述待下刷数据写入缓存中。

[0070] 由于普通数据存在读操作,因此需要将普通数据写入缓存,以提高存储系统的读性能。在本步骤中,对于普通数据可以采用写穿策略,也可以采用写回策略,在此不进行限定。若普通数据的写策略为写穿策略,本步骤包括:将所述待下刷数据同时写入所述缓存和所述硬盘驱动器中。若普通数据的写策略为写回策略,则本步骤包括:将所述待下刷数据写入缓存中;当满足预设条件时,将所述缓存中的数据刷新至所述硬盘驱动器中。此处的预设条件可以为读取命中率小于预设值,即当缓存的读取命中率小于预设值时,将缓存中读取命中率最低的数据块中的数据刷新至硬盘驱动器。预设条件也可以为缓存已满,即当缓存已满时,将缓存中的部分数据刷新至硬盘驱动器,当然还可以采用其他策略将缓存中的数据刷新至硬盘驱动器,在此不进行具体限定。

[0071] 在本申请实施例中,将客户端产生的数据划分为不同的类型,即只写数据和普通数据。只写数据在一定时间窗口内只存在写操作,而不存在读操作,也就是说,客户端的读请求在该时间窗口内不会命中只写数据,将只写数据加载到缓存,不但不会提高缓存的读取命中率,还会对缓存造成大量非必要的写入流量。因此,本申请实施例提供的数据写入方法,对只写数据和普通数据采取不同的写入方式,对于普通数据采用写穿策略或写回策略,将其写入缓存。而对于只写数据采用绕写策略,即将其直接写入后端的硬盘驱动器,可以有效减少写入缓存的无用流量,同时在缓存中为普通数据留出更多空间,提高缓存空间利用率,从而提高缓存的读取命中率,提升存储系统的读性能。由此可见,本申请实施例提供的

数据写入方法,实现了对缓存的最低写入流量,提高了写策略的效率。

[0072] 本实施例基于统计学确定待下刷数据的类型,将以存储服务器为执行主体进行介绍。具体的:

[0073] 参见图3,本申请实施例提供的第二种数据写入方法的流程图,如图3所示,包括:

[0074] S201:当接收到写入请求时,将所述写入请求对应的写入数据写入写缓冲区中;

[0075] S202:当触发所述写缓冲区的数据下刷操作时,获取所述待下刷数据对应的数据块的历史访问数据;

[0076] S203:基于所述历史访问数据利用统计学方法判断待下刷数据是否为只写数据;若是,则进入S204;若否,则进入S205;

[0077] 本实施例基于历史访问数据利用统计学方法确定待下刷数据的类型,此处的历史访问数据为待下刷数据对应的数据块的历史访问数据。客户端在产生写入请求时,可以指定写入地址,此处的数据块即为该写入地址对应的数据块。历史访问数据描述该数据块的历史访问情况,可以包括在一定时间窗口内的写请求次数、读请求次数等,依据历史访问数据可以确定该数据块存储只写数据的概率,进而确定即将存储在该数据块内的待下刷数据是否为只写数据。

[0078] 作为一种优选实施方式,本步骤包括:基于所述历史访问数据确定多个时间窗口内所述待下刷数据对应的数据块的写请求次数,并根据所有所述写请求次数计算所述待下刷数据对应的数据块的只写概率;其中,所述只写概率描述所述数据块存储只写数据的概率;判断所述只写概率是否大于第三预设值,若是,则判定所述待下刷数据为所述只写数据,若否,则判定所述待下刷数据为所述普通数据。

[0079] 在具体实施中,基于历史访问数据确定多个时间窗口内待下刷数据对应的数据块的写请求次数和总请求次数,此处不对时间窗口进行限定,例如一个时间窗口可以为1天,多个时间窗口可以为一个月,即确定一个月内每天中该数据块的写请求次数和总请求次数。总请求次数既包括写请求次数也包括读请求次数,计算每个时间窗口内该数据块的写请求比例,即写请求次数与总请求次数的比值,并将所有时间窗口的写请求比例的平均值作为该数据块的只写概率,该只写概率可以表征该数据块存储只写数据的概率。当只写概率大于第三预设值时,可以判定该数据块即将存储的数据为只写数据,即待下刷数据为只写数据,否则判定待下刷数据为普通数据。

[0080] S204:将所述待下刷数据写入硬盘驱动器中;

[0081] S205:将所述待下刷数据写入缓存中。

[0082] 由此可见,本实施例基于待下刷数据对应的数据块的历史访问数据,利用统计学方法确定待下刷数据的类型,对只写数据和普通数据采取不同的写入方式,对于普通数据采用写穿策略或写回策略,将其写入缓存。而对于只写数据采用绕写策略,即将其直接写入后端的硬盘驱动器,可以有效减少写入缓存的无用流量,同时在缓存中为普通数据留出更多空间,提高缓存空间利用率,从而提高缓存的读取命中率,提升存储系统的读性能。

[0083] 本实施例利用分类器确定待下刷数据的类型,同样将以存储服务器为执行主体进行介绍。具体的:

[0084] 参见图4,本申请实施例提供的第三种数据写入方法的流程图,如图4所示,包括:

[0085] S301:当接收到写入请求时,将所述写入请求对应的写入数据写入写缓冲区中;

[0086] S302:当触发所述写缓冲区的数据下刷操作时,获取所述待下刷数据对应的数据块的历史访问数据,并在所述历史访问数据中提取所述待下刷数据对应的数据块的特征信息作为第一特征信息;

[0087] 本实施例利用分类器确定待下刷数据的类型,在本步骤中,首先基于待下刷数据对应的数据块的历史访问数据提取待下刷数据的特征信息作为第一特征信息,此处的特征信息可以包括时间特征和请求类特征中的任一项或两项,时间特征为描述访问数据块的时间的特征,请求类特征为描述访问数据块的请求的特征。

[0088] 具体的,时间特征可以包括上次访问时间戳和平均重用时间差中的任一项或两项,上次访问时间戳描述上一次访问数据块的时间点,平均重用时间差描述访问数据块的平均时间间隔。可以理解的是,时间特征可以表示数据块的访问新进度和时间间隔,即上次访问时间戳描述访问新进度,平均重用时间差描述访问时间间隔,当然也可以使用当前时间和上一次访问数据块的时间点的时间间隔表示数据块的访问新进度,在此不进行具体限定。另外,可以对上述时间特征进行标准化处理,例如,将时间特征的单位设置为小时,并将数值的上限限制为100。

[0089] 可以理解的是,本实施例默认存在训练分类器的步骤,在具体实施中,可以利用机器学习算法进行训练,而机器学习算法中的常用特征信息还可以使用空间特征,即待写入数据的地址信息,如卷ID、偏移等。但是,地址会随着时间而变化,影响分类器的预测,因此在本实施例中,不使用空间特征,而采用请求类特征来丰富训练特征。

[0090] 具体的,请求类特征可以包括平均请求大小、大请求比率、小请求比率和写请求比率中的任一项或任几项的组合,大请求比率描述访问数据块的请求中请求大小大于第一预设值的请求的比率,小请求比率描述访问数据块的请求中请求大小小于第二预设值的请求的比率,写请求比率描述访问数据块的请求中写请求的比率,所述第一预设值大于或等于所述第二预设值。

[0091] 例如,平均请求大小是访问数据块的平均请求大小,单位可以为千字节,上限可以为100KB。大请求比率可以为请求大小大于64KB的请求的比率,小请求比率是请求大小小于8KB的请求的比率。

[0092] 需要说明的是,若对每个数据块统计特征信息,特征信息的提取会消耗过多的系统资源,存储过程中的内存占用过大,导致特征信息提取效率较低。因此,为了提高特征信息的提取效率,可以提高特征信息的统计粒度,然而过大的统计粒度也会导致训练完成的分类器预测结果不准确。因此,可以以数据区域作为特征信息的统计粒度,每个数据区域包括多个数据块,以在内存占用和预测准确性之间进行权衡。存储系统中的访问粒度,即最小读写粒度为数据块,例如8KB,数据区域的大小为访问粒度的整数倍,例如数据区域的大小可以为1MB。在具体实施中,以1MB的统计粒度统计特征信息连续数据区域进行统计,每个数据区域的特征信息来自该数据区域中所有数据块的历史读取数据,同一数据区域中的数据块共享此数据区域的特征信息。即所述获取所述待下刷数据对应的数据块的特征信息作为第一特征信息的步骤包括:确定所述待下刷数据对应的数据块所属的数据区域,提取该数据区域的特征信息作为第一特征信息。

[0093] S303:将所述第一特征信息输入训练完成的分类器中以便判断待下刷数据是否为只写数据;若是,则进入S304;若否,则进入S305;

[0094] 在本步骤中,将上一步骤提取的第一特征信息输入训练完成的分类器中,得到待下刷数据的类型,若其为只写数据,则进入S304,若其为普通数据,则进入S305。

[0095] S304:将所述待下刷数据写入硬盘驱动器中;

[0096] S305:将所述待下刷数据写入缓存中。

[0097] 由此可见,本实施例利用分类器确定待下刷数据的类型,分类器的预测准确度较高,提高了确定数据类型的准确度。从写缓冲区中下刷的待写入数据均首先经过分类器分类,然后根据分类结果进行不同操作,即对普通数据采用写回策略或写穿策略,对只写数据采用绕写策略。通过上述方式极大地减少了对缓存的写流量,同时也为普通数据保留了更多的缓存空间,提高缓存空间的资源利用率,从而提高了存储系统的读性能。

[0098] 本实施例基于机器学习算法对分类器进行训练,机器学习(英文全称:Machine Learning,英文简称:ML)是一门多领域交叉学科,涉及概率论、统计学、逼近论、凸分析、算法复杂度理论等多门学科。专门研究计算机怎样模拟或实现人类的学习行为,以获取新的知识或技能,重新组织已有的知识结构使之不断改善自身的性能。机器学习是人工智能的核心,是使计算机具有智能的根本途径,其应用遍及人工智能的各个领域。机器学习和深度学习通常包括人工神经网络、置信网络、强化学习、迁移学习、归纳学习、式教学习等技术。具体的:

[0099] 参见图5,本申请实施例提供一种分类器训练方法的流程图,如图5所示,包括:

[0100] S401:当所述缓存触发数据替换操作时,基于替换数据的历史读取数据对所述替换数据进行标记;

[0101] 本实施例利用机器学习算法对分类器进行训练,需要说明的是,此处不对具体的机器学习算法进行限定,可以使用监督机器学习算法,例如朴素贝叶斯(Naive Bayes)、Logistic回归、决策树、AdaBoost和随机森林等。在X86-64计算服务器上运行分类器,该服务器有两个Intel Xeon E5-2670-v3CPU和128GB DRAM内存,不同机器学习算法的分类结果如表1所示,表1中的训练时间为训练一天的训练样本所用的时间。

[0102] 表1

[0103]

分类器名称	准确率(%)	召回率(%)	训练时间(ms)	预测时间(ms)
Logistic回归	83.54	75.21	500.84	1.01
朴素贝叶斯	83.89	83.91	440.88	1.25
决策树	80.96	68.61	482.62	1.16
AdaBoost	87.85	85.64	462.47	14.73
随机森林	88.32	82.64	523.68	20.23

[0104] 由表1可知,随机森林在准确率方面优于其他算法,但需要更多的时间进行预测,每个请求平均需要20.23 μ s。AdaBoost在准确率方面仅次于随机森林,并且具有较短的预测时间,每个请求平均为14.73 μ s。朴素贝叶斯能很好的平衡准确率、召回率和预测时间,Logistic回归在召回率方面不如朴素贝叶斯,决策树具有最低的准确率和召回率。

[0105] 在具体实施中,在一定的时间窗口内采集训练样本用于训练分类器,例如,每天采集训练样本对分类器进行训练,训练完成的分类器用于第二天数据类型的区分。所有监督型机器学习算法训练一天样本耗时几百毫秒,分类器模型的大小仅为几兆字节,可见训练分类器对存储系统的影响可以忽略不计。

[0106] 在本实施例中,训练样本为替换数据,即当缓存触发数据替换操作时,将缓存中的第一数据替换为硬盘驱动器中的第二数据时,第一数据即为本步骤中的替换数据。基于其历史读取数据对其进行标记,例如,若该替换数据具有一个或多个读取命中次数,则将其标记为0(0表示普通数据),否则将其标记为1(1表示只写数据)。

[0107] S402:获取所述替换数据对应的数据块的特征信息作为第二特征信息,并利用标记完成的替换数据和所述第二特征信息训练分类器得到所述训练完成的分类器。

[0108] 在本步骤中,提取训练样本的特征信息作为第二特征信息,此处的第二特征信息也可以包括时间特征、请求类特征,在此不再赘述。将标记完成的训练样本和训练样本的特征信息输入分类器中,用于对该分类器进行训练,得到训练完成的分类器。训练完成的分类器用于确定数据的类型,只写数据或普通数据。

[0109] 可以理解的是,为了提高训练样本的特征信息的提取效率,此处也可以将数据区域作为训练样本的特征信息的统计粒度。即所述获取所述替换数据对应的数据块的特征信息作为第二特征信息的步骤包括:确定所述替换数据对应的数据块所属的数据区域;提取所述数据区域的特征信息作为所述第二特征信息。

[0110] 由此可见,本实施例利用机器学习算法对分类器进行训练,在提取训练样本的特征信息时,使用请求类特征丰富分类器的训练特征,使得分类器的分类效果更好,提高了确定数据类型的准确度。

[0111] 本申请实施例公开了一种数据写入方法,相对于前几个实施例,本实施例对技术方案作了进一步的说明和优化,同样以存储服务器为执行主体进行介绍。具体的:

[0112] 参见图6,本申请实施例提供的第四种数据写入方法的流程图,如图6所示,包括:

[0113] S501:当接收到写入请求时,将所述写入请求对应的写入数据写入写缓冲区中;

[0114] S502:当所述写缓冲区的数据量达到阈值时,触发写缓冲区的数据下刷操作;

[0115] S503:获取所述待下刷数据对应的数据块的历史访问数据,并在所述历史访问数据中提取所述待下刷数据对应的数据块的特征信息作为第一特征信息;

[0116] S504:将所述第一特征信息输入训练完成的分类器中以便判断待下刷数据是否为只写数据;若是,则进入S505;若否,则进入S507;

[0117] S505:将所述固态硬盘中所述待下刷数据对应的脏数据刷新至所述硬盘驱动器;

[0118] S506:将所述待下刷数据写入硬盘驱动器中;

[0119] S507:将所述待下刷数据写入缓存中;

[0120] S508:当满足预设条件时,将所述缓存中的数据刷新至所述硬盘驱动器中。

[0121] 由此可见,本实施例提供了一种基于机器学习的写策略,可以准确识别不同类型的数据,即只写数据和普通数据,进而对不同类型的数据进行不同的处理。对普通数据采用写回策略,对只写数据采用绕写策略,最大程度的减少了对缓存的写流量,同时也为普通数据保留了更多的缓存空间,提高缓存空间的资源利用率,从而提高了存储系统的读性能。

[0122] 为了便于理解,结合本申请的一种应用场景进行介绍。具体的,如图7所示,数据写入系统包括客户端、分类器和云块存储产品腾讯CBS的存储区。客户端中设置Hypervisor(又称虚拟机监视器,英文全称:virtual machine monitor,英文简称:VMM),用于监控客户端的写入请求。分类器采集训练样本,用于训练机器学习模型。存储区包括写缓冲区和后端

存储,后端存储包括SSD缓存和HDD,写策略决定是否将待写入数据加载到SSD缓存以及何时将SSD缓存中的脏数据下刷到HDD。如图8所示,当存在来自客户端的请求时,写策略的处理流程如下:

[0123] 步骤一:将所有写入请求写入写缓冲区并顺序记录到日志文件中,在脏数据量达到阈值时通知存储服务器。

[0124] 步骤二:当写缓冲区下刷数据时,分类器获取当前刷新数据块的所在区域的特征并分析其类型。如果当前刷新数据块存储的数据为只写数据,则执行步骤三,如果是普通数据,则执行步骤四。

[0125] 步骤三:查找当前刷新数据块在SSD缓存中是否存在脏数据,若存在,则首先将SSD缓存中的脏数据下刷,写缓冲区中的下刷数据则直接写入HDD,流程结束。

[0126] 步骤四:将下刷数据写入SSD缓存,然后采用写回策略异步刷新到HDD。此处使用写回策略而不是写穿策略的原因是SSD具有非易失性的优势,因此写回策略足以确保数据一致性。

[0127] 其中,分类器每天训练一次模型以进行第二天的预测。在系统运行过程中收集训练样本。在SSD缓存存在需要剔除数据时,将剔除数据下刷到HDD,并添加一个训练样本。

[0128] 实验结果如表2所示:

[0129] 表2

[0130]

写策略	命中率	SSD写入流量	读取延迟
绕写策略	92.13%	1	1.29ms
写回策略	94.68%	7.49	934.16μs
本实施例的写策略	97.22%	4.38	583.64μs

[0131] 在表2中,其他策略的写入流量被标准化为写入策略的写入流量。绕写策略实现了对SSD的最低写入流量(1),这是因为错误写入数据首先直接写入后端存储,然而,绕写策略获得最低命中率92.13%,并且引起最高读取延迟1.29ms。可见,绕写策略虽然实现了最低的SSD写入流量,但在命中率和读取延迟方面破坏了缓存性能。写回策略导致SSD的最大写入流量(7.49)。本实施例提供的写策略与写回策略相比,将命中率从94.68%提高到97.22%,提高了2.61%,并将读取延迟和写入流量减少了37.52%(934.16μs到583.64μs)和41.52%(从7.49到4.38)。由此可见,本实施例提供的写策略实现了最佳性能。

[0132] 下面对本申请实施例提供的一种数据写入装置进行介绍,下文描述的一种数据写入装置与上文描述的一种数据写入方法可以相互参照。

[0133] 参见图9,本申请实施例提供的一种数据写入装置的结构图,如图9所示,包括:

[0134] 第一写入模块901,用于当接收到写入请求时,将所述写入请求对应的写入数据写入写缓冲区中;

[0135] 获取模块902,用于当触发所述写缓冲区的数据下刷操作时,获取所述待下刷数据对应的数据块的历史访问数据;

[0136] 判断模块903,用于基于所述历史访问数据判断待下刷数据是否为只写数据;

[0137] 第二写入模块904,用于当所述待下刷数据为只写数据时,将所述待下刷数据写入硬盘驱动器中;

[0138] 第三写入模块905,用于当所述待下刷数据为非只写数据时,将所述待下刷数据写

入缓存中。

[0139] 在本申请实施例中,将客户端产生的数据划分为不同的类型,即只写数据和普通数据。只写数据在一定时间窗口内只存在写操作,而不存在读操作,也就是说,客户端的读请求在该时间窗口内不会命中只写数据,将只写数据加载到缓存,不但不会提高缓存的读取命中率,还会对缓存造成大量非必要的写入流量。因此,本申请实施例提供的数据写入装置,对只写数据和普通数据采取不同的写入方式,对于普通数据采用写穿策略或写回策略,将其写入缓存。而对于只写数据采用绕写策略,即将其直接写入后端的硬盘驱动器,可以有效减少写入缓存的无用流量,同时在缓存中为普通数据留出更多空间,提高缓存空间利用率,从而提高缓存的读取命中率,提升存储系统的读性能。由此可见,本申请实施例提供的数据写入装置,实现了对缓存的最低写入流量,提高了写策略的效率。

[0140] 在上述实施例的基础上,作为一种优选实施方式,所述第二写入模块904包括:

[0141] 第一刷新单元,用于将所述缓存中所述待下刷数据对应的脏数据刷新至所述硬盘驱动器;

[0142] 第一写入单元,用于将所述待下刷数据写入硬盘驱动器中。

[0143] 在上述实施例的基础上,作为一种优选实施方式,所述第三写入模块905包括:

[0144] 第二写入单元,用于将所述待下刷数据写入缓存中;

[0145] 第二刷新单元,用于当满足预设条件时,将所述缓存中的数据刷新至所述硬盘驱动器中。

[0146] 在上述实施例的基础上,作为一种优选实施方式,所述第三写入模块905具体为将所述待下刷数据同时写入所述缓存和所述硬盘驱动器中的模块。

[0147] 在上述实施例的基础上,作为一种优选实施方式,所述获取模块902具体为当所述写缓冲区的数据量达到阈值时,触发写缓冲区的数据下刷操作,并获取所述待下刷数据对应的数据块的历史访问数据。

[0148] 在上述实施例的基础上,作为一种优选实施方式,所述判断模块903具体为在所述历史访问数据中提取所述待下刷数据对应的数据块的特征信息作为第一特征信息,将所述第一特征信息输入训练完成的分类器中以便判断待下刷数据是否为只写数据的模块。

[0149] 在上述实施例的基础上,作为一种优选实施方式,还包括:

[0150] 标记模块,用于当所述缓存触发数据替换操作时,基于替换数据的历史读取数据对所述替换数据进行标记;

[0151] 训练模块,用于获取所述替换数据对应的数据块的特征信息作为第二特征信息,并利用标记完成的替换数据和所述第二特征信息训练分类器得到所述训练完成的分类器。

[0152] 在上述实施例的基础上,作为一种优选实施方式,所述训练模块包括:

[0153] 确定单元,用于确定所述替换数据对应的数据块所属的数据区域;其中,所述数据区域包括多个数据块,所述数据区域的大小为访问粒度的整数倍;

[0154] 提取单元,用于提取所述数据区域的特征信息作为所述第二特征信息;

[0155] 训练单元,用于利用标记完成的替换数据和所述第二特征信息训练分类器得到所述训练完成的分类器。

[0156] 在上述实施例的基础上,作为一种优选实施方式,所述特征信息包括时间特征和/或请求类特征;所述时间特征为描述访问数据块的时间的特征,所述请求类特征为描述访

问数据块的请求的特征。

[0157] 在上述实施例的基础上,作为一种优选实施方式,所述时间特征包括上次访问时间戳和/或平均重用时间差;所述上次访问时间戳描述上一次访问数据块的时间点,所述平均重用时间差描述访问数据块的平均时间间隔;

[0158] 所述请求类特征包括平均请求大小、大请求比率、小请求比率和写请求比率中的任一项或任几项的组合;所述大请求比率描述访问数据块的请求中请求大小大于第一预设值的请求的比率,所述小请求比率描述访问数据块的请求中请求大小小于第二预设值的请求的比率,所述写请求比率描述访问数据块的请求中写请求的比率,所述第一预设值大于或等于所述第二预设值。

[0159] 在上述实施例的基础上,作为一种优选实施方式,所述判断模块903具体为基于所述历史访问数据利用统计学方法判断待下刷数据是否为只写数据的模块。

[0160] 在上述实施例的基础上,作为一种优选实施方式,所述判断模块903包括:

[0161] 计算单元,用于基于所述历史访问数据确定多个时间窗口内所述待下刷数据对应的数据块的写请求次数,并根据所有所述写请求次数计算所述待下刷数据对应的数据块的只写概率;其中,所述只写概率描述所述数据块存储只写数据的概率;

[0162] 判断单元,用于判断所述只写概率是否大于第三预设值,若是,则判定所述待下刷数据为所述只写数据,若否,则判定所述待下刷数据为所述普通数据。

[0163] 关于上述实施例中的装置,其中各个模块执行操作的具体方式已经在有关该方法的实施例中进行了详细描述,此处将不做详细阐述说明。

[0164] 本申请还提供了一种存储服务器,参见图10,本申请实施例提供的一种存储服务器200的结构图,如图10所示,可以包括处理器21和存储器22。

[0165] 其中,处理器21可以包括多个处理核心,比如4核心处理器、8核心处理器等。处理器21可以采用DSP (Digital Signal Processing, 数字信号处理)、FPGA (Field-Programmable Gate Array, 现场可编程门阵列)、PLA (Programmable Logic Array, 可编程逻辑阵列) 中的至少一种硬件形式来实现。处理器21也可以包括主处理器和协处理器,主处理器是用于对在唤醒状态下的数据进行处理的处理单元,也称CPU (Central Processing Unit, 中央处理器);协处理器是用于对在待机状态下的数据进行处理的低功耗处理器。在一些实施例中,处理器21可以在集成有GPU (Graphics Processing Unit, 图像处理器), GPU 用于负责显示屏所需要显示的内容的渲染和绘制。一些实施例中,处理器21还可以包括AI (Artificial Intelligence, 人工智能) 处理器,该AI处理器用于处理有关机器学习的计算操作。

[0166] 存储器22可以包括多个计算机可读存储介质,该计算机可读存储介质可以是非暂态的。存储器22还可包括高速随机存取存储器,以及非易失性存储器,比如多个磁盘存储设备、闪存存储设备。本实施例中,存储器22至少用于存储以下计算机程序221,其中,该计算机程序被处理器21加载并执行之后,能够实现前述任一实施例公开的由服务器侧执行的数据写入方法中的相关步骤。另外,存储器22所存储的资源还可以包括操作系统222和数据223等,存储方式可以是短暂存储或者永久存储。其中,操作系统222可以包括Windows、Unix、Linux等。

[0167] 在一些实施例中,存储服务器20还可包括有显示屏23、输入输出接口24、通信接口

25、传感器22、电源27以及通信总线28。

[0168] 当然,图10所示的存储服务器的结构并不构成对本申请实施例中存储服务器的限定,在实际应用中存储服务器可以包括比图10所示的更多或更少的部件,或者组合某些部件。

[0169] 在另一示例性实施例中,还提供了一种包括程序指令的计算机可读存储介质,该程序指令被处理器执行时实现上述任一实施例存储服务器所执行的数据写入方法的步骤。

[0170] 说明书中各个实施例采用递进的方式描述,每个实施例重点说明的都是与其他实施例的不同之处,各个实施例之间相同相似部分互相参见即可。对于实施例公开的装置而言,由于其与实施例公开的方法相对应,所以描述的比较简单,相关之处参见方法部分说明即可。应当指出,对于本技术领域的普通技术人员来说,在不脱离本申请原理的前提下,还可以对本申请进行若干改进和修饰,这些改进和修饰也落入本申请权利要求的保护范围内。

[0171] 还需要说明的是,在本说明书中,诸如第一和第二等之类的关系术语仅仅用来将一个实体或者操作与另一个实体或操作区分开来,而不一定要求或者暗示这些实体或操作之间存在任何这种实际的关系或者顺序。而且,术语“包括”、“包含”或者其任何其他变体意在涵盖非排他性的包含,从而使得包括一系列要素的过程、方法、物品或者设备不仅包括那些要素,而且还包括没有明确列出的其他要素,或者是还包括为这种过程、方法、物品或者设备所固有的要素。在没有更多限制的情况下,由语句“包括一个……”限定的要素,并不排除在包括所述要素的过程、方法、物品或者设备中还存在另外的相同要素。

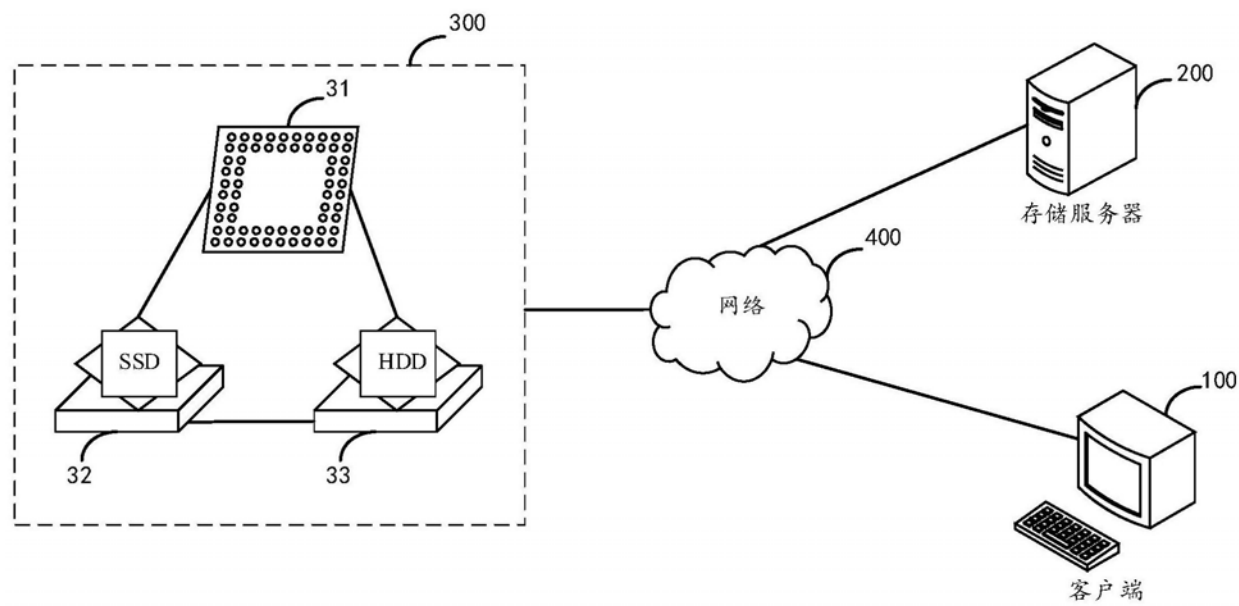


图1

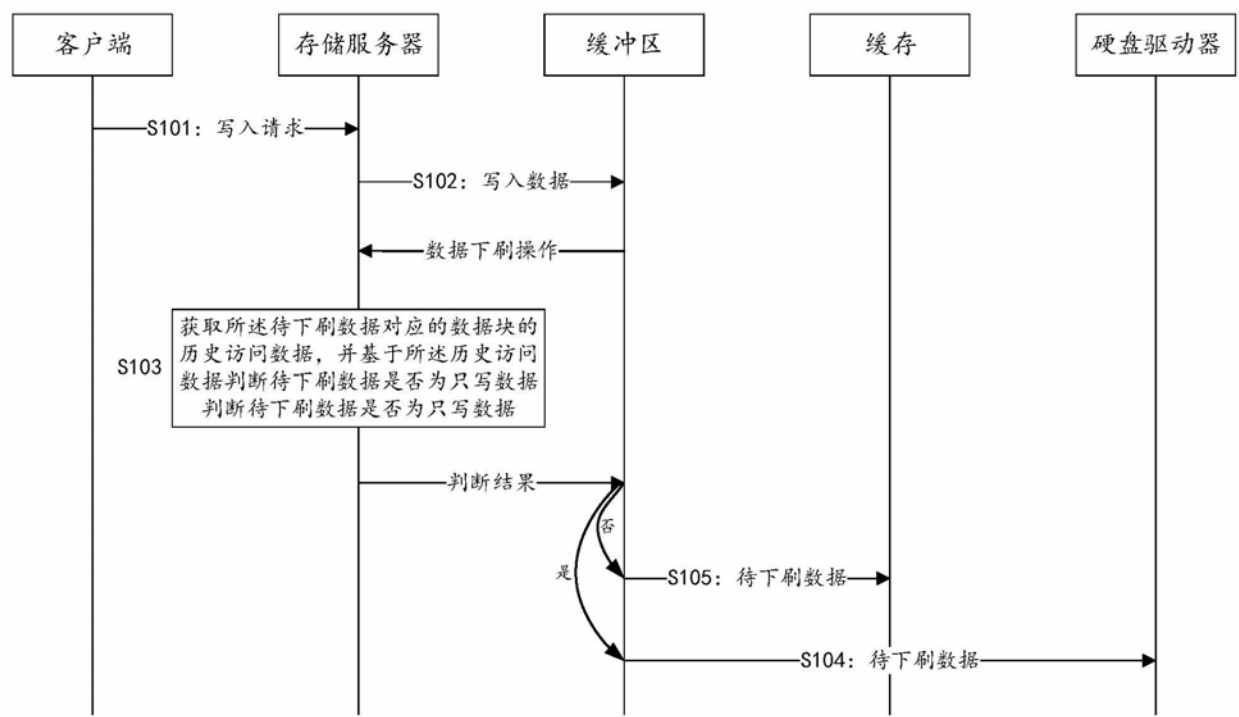


图2

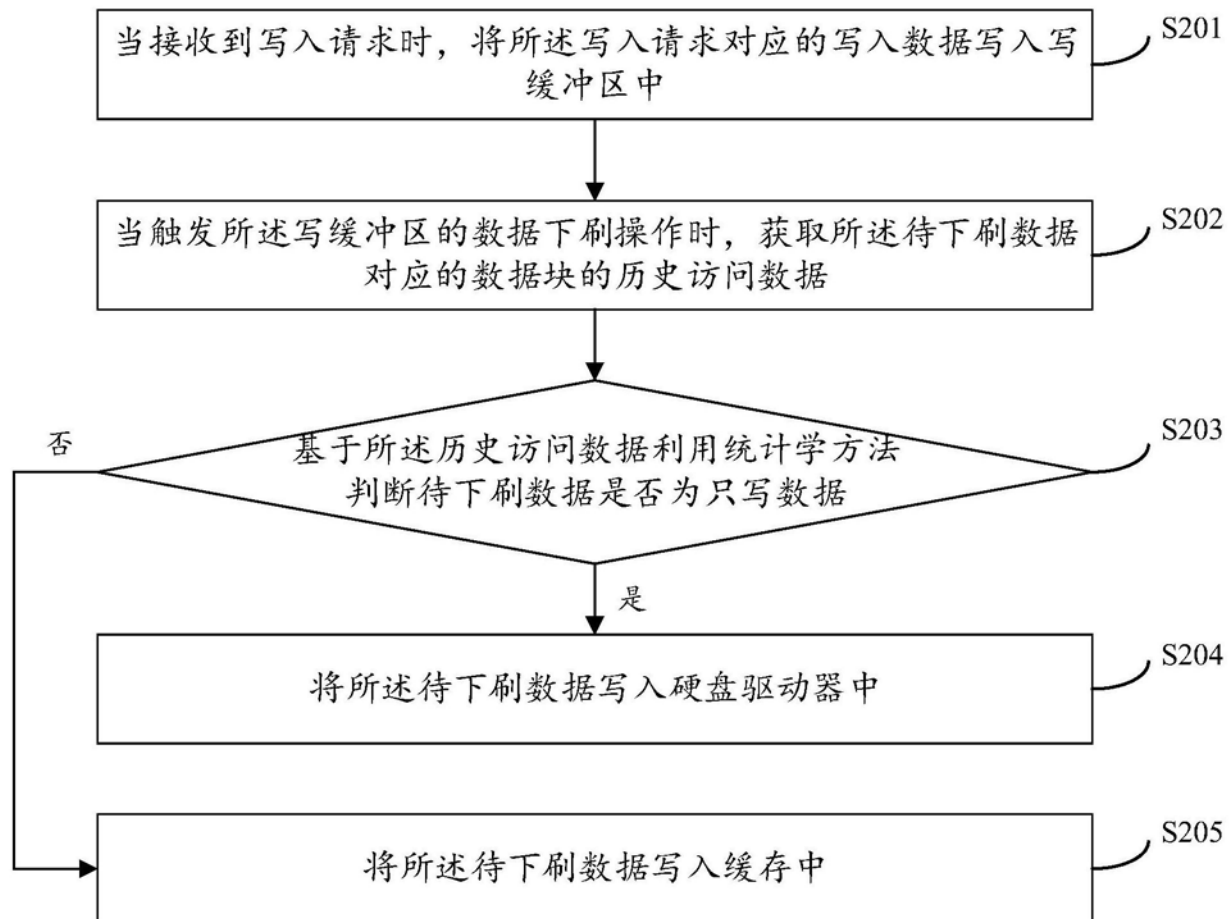


图3

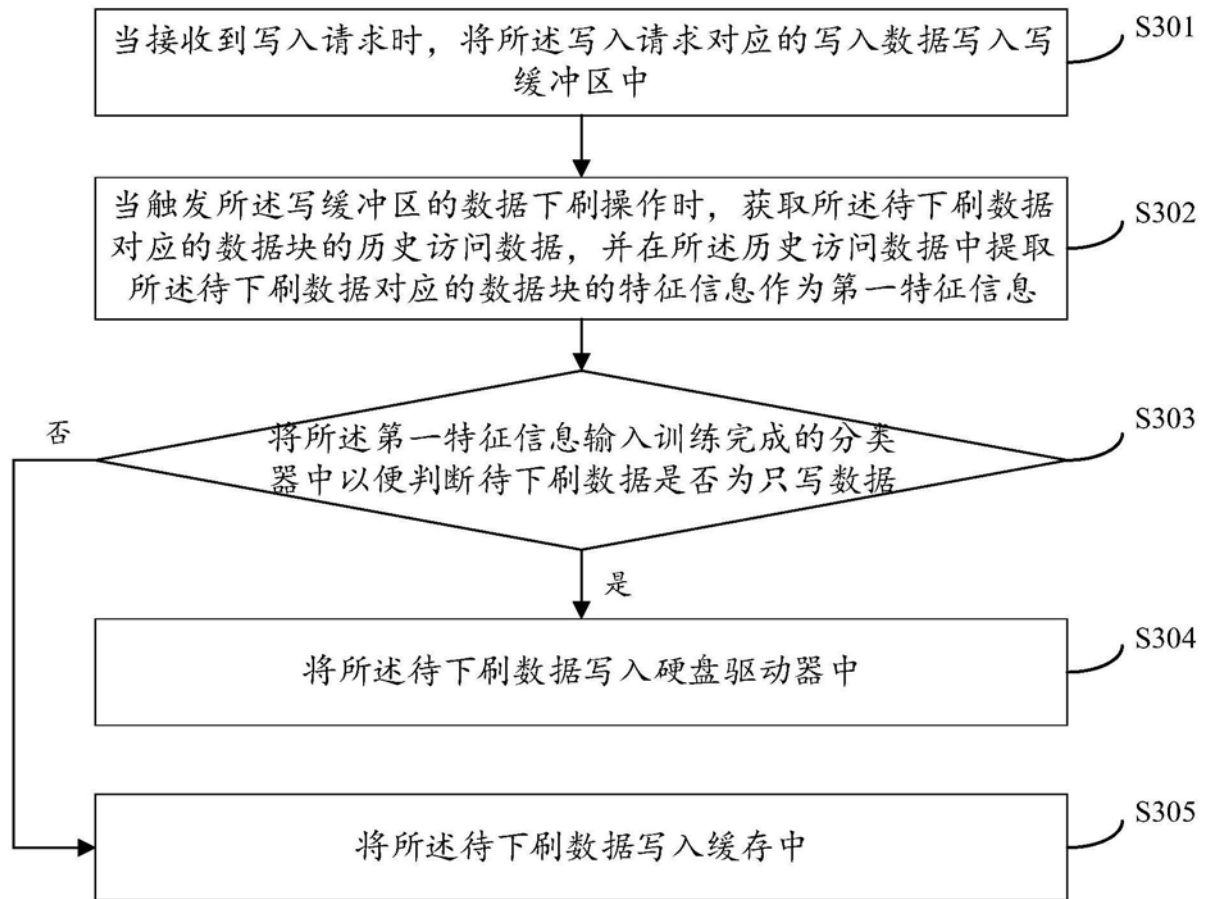


图4

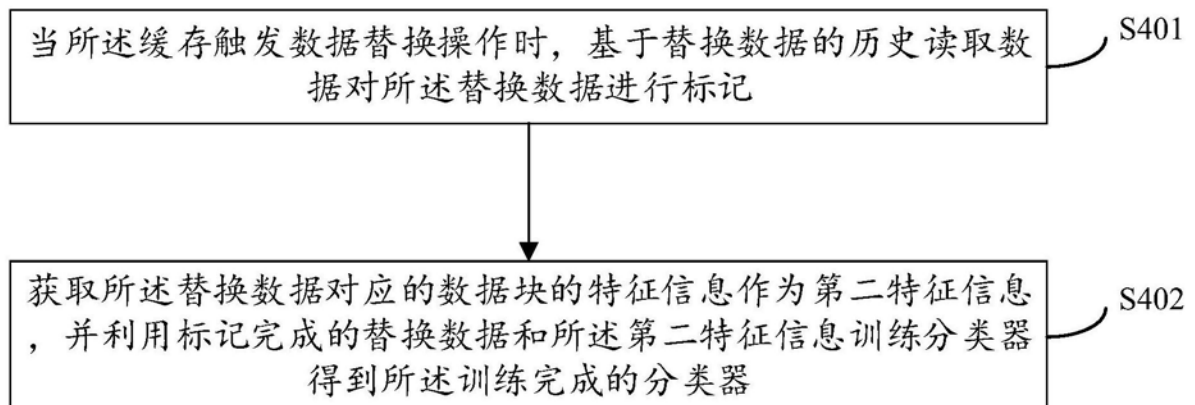


图5

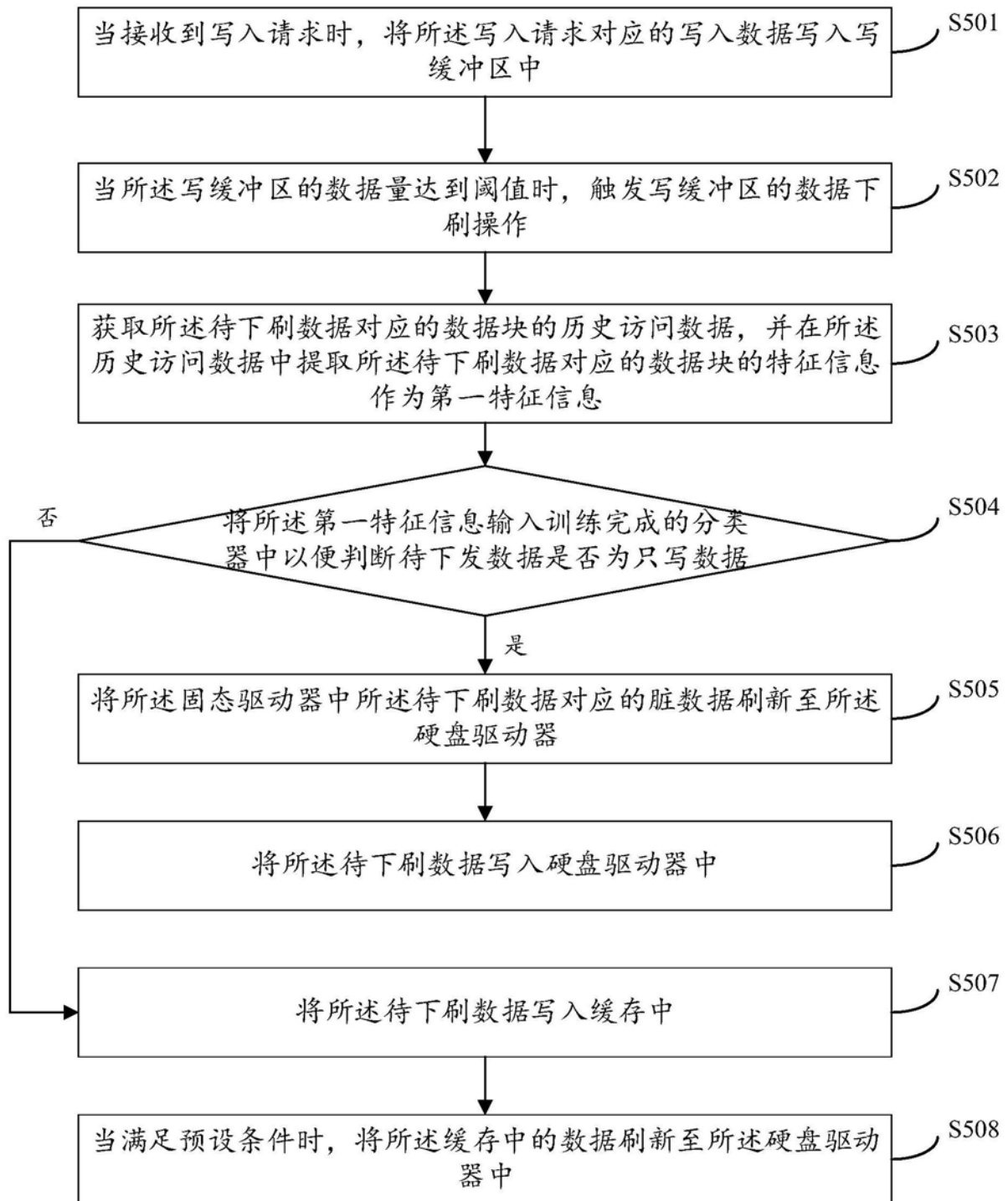


图6

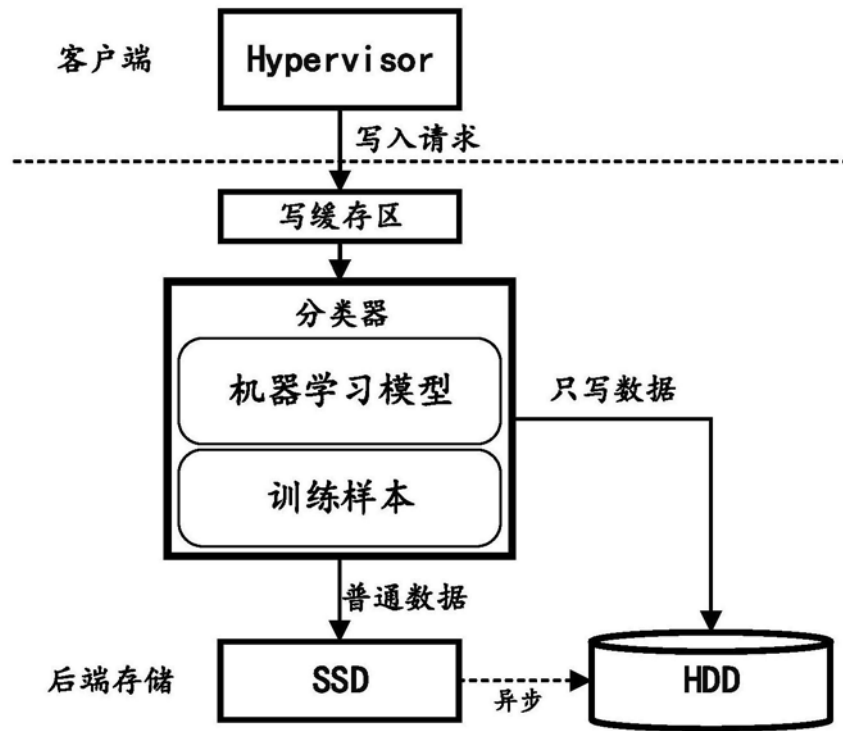


图7

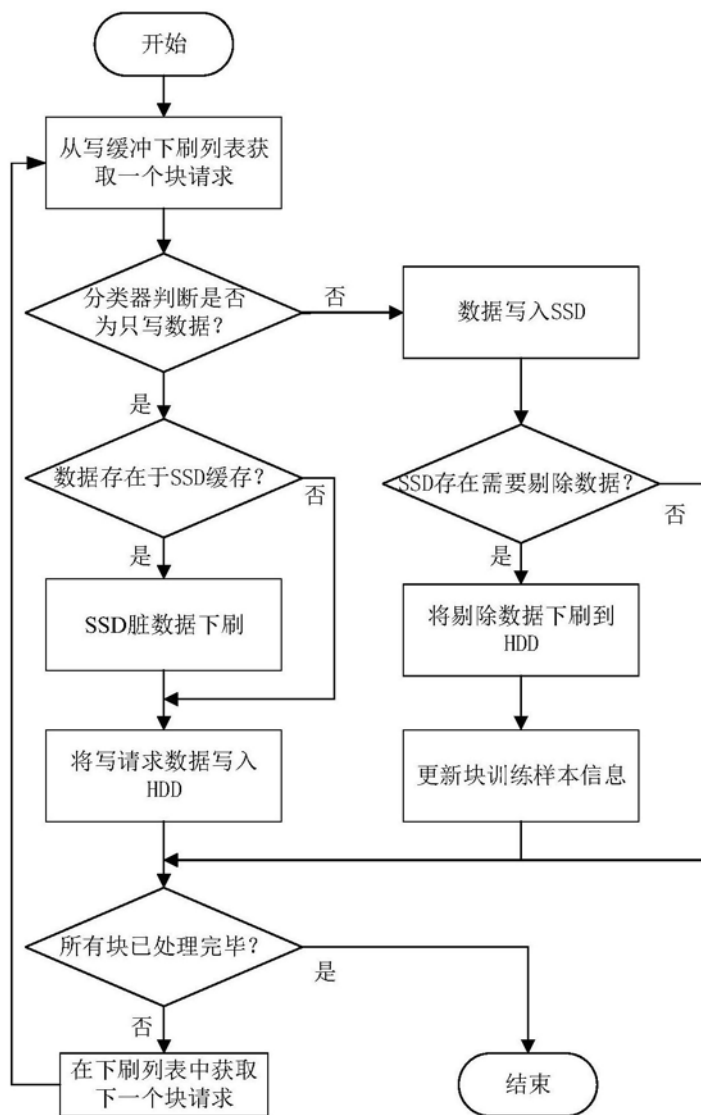


图8

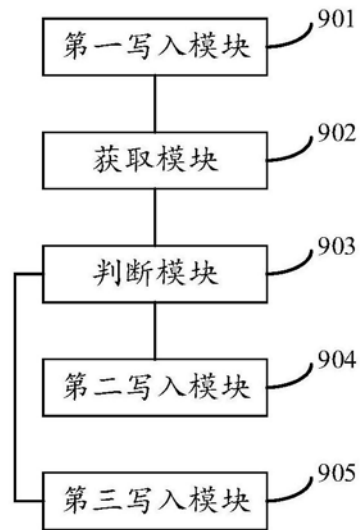


图9

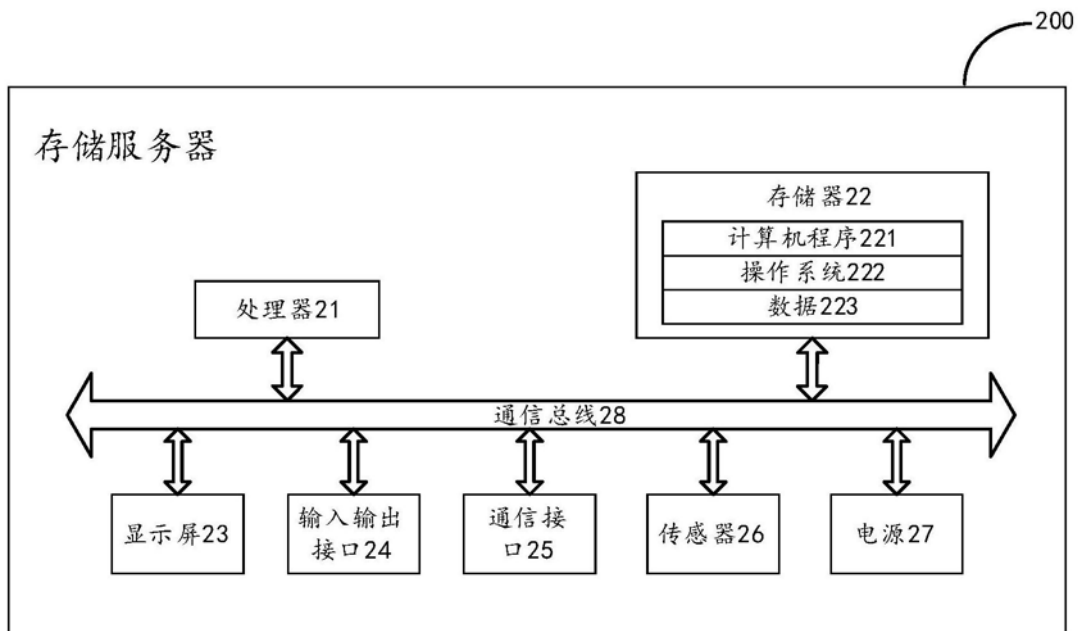


图10