



(12) 发明专利

(10) 授权公告号 CN 113760186 B

(45) 授权公告日 2025. 01. 14

(21) 申请号 202110857597.1

(56) 对比文件

(22) 申请日 2019.07.30

CN 101166273 A, 2008.04.23

(65) 同一申请的已公布的文献号

审查员 罗朋

申请公布号 CN 113760186 A

(43) 申请公布日 2021.12.07

(62) 分案原申请数据

201910697302.1 2019.07.30

(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

(72) 发明人 何昌军 陈灿

(51) Int. Cl.

G06F 3/06 (2006.01)

G06F 12/0868 (2016.01)

G06F 11/10 (2006.01)

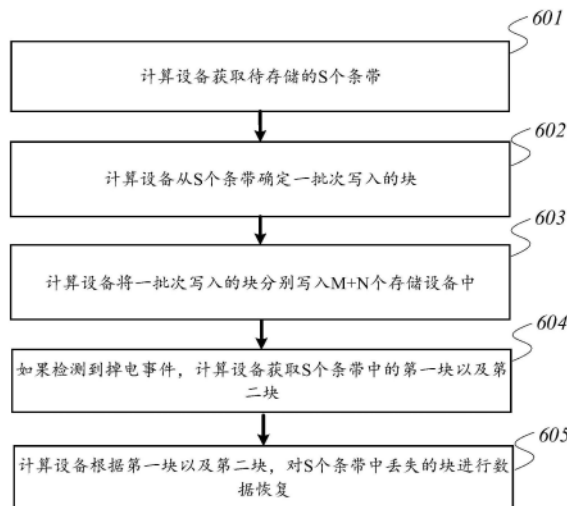
权利要求书2页 说明书20页 附图10页

(54) 发明名称

数据存储方法、装置、计算设备、存储系统及存储介质

(57) 摘要

本申请提供了一种数据存储方法、装置、计算设备、存储系统及存储介质,属于存储技术领域。本申请对条带中的块写入至存储设备的先后顺序进行改进,从将一批次写入同一条带的M+N个块,改进为一批次写入同一条带的小于或等于N个块,对于每个条带而言,由于将该条带同时写入到M+N个存储设备的块的数量小于或等于N个,那么条带同时缓存至M+N个存储设备的块的数量会小于或等于N个,因此即使M+N个存储设备在任一时间点全部掉电,且存储设备没有备份电源,使得M+N个存储设备缓存的数据全部丢失,该条带中丢失的块的数量会小于或等于N个,因此条带可以通过冗余算法恢复出丢失的块,从而避免丢失条带,保证数据存储的可靠性。



1. 一种数据存储方法,应用于计算设备,其特征在于,所述方法包括:

所述计算设备从多个条带中确定需要同一批次写入到存储系统的块;其中每一个条带均包含M个数据块和N个校验块;所述同一批次写入到存储设备的块中来自同一个条带的块的数量小于或等于N;其中M和N均为正整数;

所述计算设备将所述需要同一批次写入到所述存储系统的块分别写入到所述存储系统中相应的存储设备,其中,当确定需要同一批次写入到存储系统的块后,将上个同一批次中的第一个块待写入的存储设备的下一个存储设备作为本次确定的第一个块待写入的存储设备,其中,所述需要同一批次写入到存储系统的块包括  $(M+N)*K$  个,K为所述存储系统中的存储设备能够缓存的块的个数,K为大于或等于1的整数。

2. 根据权利要求1所述的方法,其特征在于,所述同一批次写入到存储系统的块的数量为M+N。

3. 根据权利要求1或2所述的方法,其特征在于,对于一个接收写入块的存储设备,相邻两次写入的块来自不同的条带。

4. 根据权利要求1或2所述的方法,其特征在于,所述将所述需要同一批次写入到所述存储系统的块分别写入到所述存储系统中相应的存储设备之后,所述方法还包括:

如果检测到掉电事件,所述计算设备获取所述多个条带中的尚未写入至所述存储系统的块以及所述存储系统中已经持久化存储的块;

所述计算设备根据所述尚未写入至所述存储系统的块以及所述已经持久化存储的块,对所述多个条带中丢失的块进行数据恢复。

5. 一种数据存储装置,其特征在于,包括:

确定模块,用于从多个条带中确定需要同一批次写入到存储系统的块;其中每一个条带均包含M个数据块和N个校验块;所述同一批次写入到存储设备的块中来自同一个条带的块的数量小于或等于N;其中M和N均为正整数;

写入模块,用于将所述需要同一批次写入到所述存储系统的块分别写入到所述存储系统中相应的存储设备,其中,当确定需要同一批次写入到存储系统的块后,将上个同一批次中的第一个块待写入的存储设备的下一个存储设备作为本次确定的第一个块待写入的存储设备,其中,所述需要同一批次写入到存储系统的块包括  $(M+N)*K$  个,K为所述存储系统中的存储设备能够缓存的块的个数,K为大于或等于1的整数。

6. 根据权利要求5所述的装置,其特征在于,所述同一批次写入到存储系统的块的数量为M+N。

7. 根据权利要求5或6所述的装置,其特征在于,对于一个接收写入块的存储设备,相邻两次写入的块来自不同的条带。

8. 根据权利要求5或6所述的装置,其特征在于,还包括:

获取模块,用于将所述需要同一批次写入到所述存储系统的块分别写入到所述存储系统中相应的存储设备之后,如果检测到掉电事件,获取所述多个条带中的尚未写入至所述存储系统的块以及所述存储系统中已经持久化存储的块;

恢复模块,用于根据所述尚未写入至所述存储系统的块以及所述已经持久化存储的块,对所述多个条带中丢失的块进行数据恢复。

9. 一种适配器,其特征在于,所述适配器包含接口和处理器;所述接口与所述处理器通

信;所述处理器用于执行权利要求1-4任一所述的方法。

## 数据存储方法、装置、计算设备、存储系统及存储介质

[0001] 本案是专利申请号为201910697302.1,申请日为2019年7月30日,发明名称为“数据存储方法、装置、计算设备、存储系统及存储介质”的分案申请。

### 技术领域

[0002] 本申请涉及存储技术领域,特别涉及一种数据存储方法、装置、计算设备、存储系统及存储介质。

### 背景技术

[0003] 随着存储技术的发展,为了提高数据存储的可靠性,数据通常采用独立磁盘冗余阵列独立磁盘冗余阵列(redundant arrays of independent drives,RAID)或纠删码(erasure code,EC)技术等方式进行冗余存储,以便通过冗余的数据恢复出丢失的数据。

[0004] 相关技术中,数据存储的流程通常包括:计算设备获取待存储的数据,将数据划分为一个或多个条带(stripe);对于一个或多个条带中的每个条带,计算设备将该条带划分为M个数据块;计算设备采用冗余算法,对M个数据块进行编码,得到N个校验块;计算设备会确定每个条带中的M+N个块(chunk),作为一批次写入的块,计算设备会按照从第一个条带至最后一个条带的顺序,将同一条带中的(M+N)个块,在同一时间点分别写入M+N个存储设备;M+N个存储设备中的每个存储设备会在同一时间点接收到同一条带中的一个块;每个存储设备会将接收到的块存入缓存;每个存储设备会对缓存的块进行持久化存储。以M为7,N为1为例,由于M+N为8,则一批次写入的块包括8个块,计算设备会先确定第一个条带中的8个块,作为第一批次写入的块;将第一个条带中的8个块,在同一时间点分别写入至8个存储设备,则第一个条带中的8个块,会在同一时间点分别缓存至8个存储设备中。计算设备会先确定第二个条带中的8个块,作为第二批次写入的块,将第二个条带中的8个块,在同一时间点分别写入至8个存储设备,则第二个条带中的8个块,也会在同一时间点分别缓存至8个存储设备中,依次类推,每执行一批次写入,会写入同一个条带中的所有块,直至将最后一个条带分别写入8个存储设备。

[0005] 其中,对于同一条带而言,如果该条带中丢失了块,当丢失的块的数量小于或等于N时,可以通过条带中剩余的块,基于冗余算法恢复出丢失的块;当丢失的块的数量大于N时,则无法恢复出丢失的块。M为大于或等于1的正整数,N为大于或等于1的正整数。

[0006] 采用上述方法存储数据时,如果M+N个存储设备同时掉电,那么M+N个存储设备缓存的数据均会丢失。而对于同一条带而言,由于该条带的每个块写入至每个存储设备的时间点相同或近似相同,会使得该条带的每个块缓存至每个存储设备的时间点相同或近似相同,那么如果缓存的时间点恰好处于掉电期间,会导致该条带的所有块均丢失。而一旦条带的所有块均丢失,就会造成该条带无法通过冗余算法恢复。为了避免这种情况造成数据丢失,存储设备中需要安装备份电源,如果存储设备掉电,备份电源会向存储设备提供电能,存储设备会利用备份电源的电能,将缓存的数据存入非易失性存储介质,而备份电源会占用存储设备的内部空间,影响存储设备的存储容量,并增加存储设备的成本。

## 发明内容

[0007] 本申请提供了一种数据存储方法、装置、计算设备、存储系统及存储介质,能够在所有存储设备同时掉电的情况下,不依赖于存储设备的备份电源,也能够避免存储设备缓存的条带丢失,从而打破存储设备必须安装备份电源的限制。

[0008] 第一方面,提供了一种数据存储方法,所述方法包括:

[0009] 从S个条带确定一批次写入的块,所述一批次写入的块中属于同一条带的块的数量小于或等于N个;其中,S个条带中的每个条带包括(M+N)个块,所述(M+N)个块包括M个数据块以及N个校验块,所述S、所述M和所述N为正整数;将所述一批次写入的块分别写入M+N个存储设备中。

[0010] 本实施例提供了一种能够在所有存储设备同时掉电的情况下,不依赖于存储设备的备份电源,也能够避免存储设备缓存的条带丢失的方案。通过对各个条带中的各个块写入至存储设备的先后顺序进行改进,从将一批次写入同一条带的M+N个块,改进为一批次写入同一条带的小于或等于N个块,那么对于每个条带而言,由于将该条带中同时写入到M+N个存储设备的块的数量小于或等于N个,那么该条带中同时缓存至M+N个存储设备的块的数量也会小于或等于N个。因此,即使M+N个存储设备在任一时间点全部掉电,造成M+N个存储设备缓存的数据全部丢失,该条带中丢失的块的数量会小于或等于N个,因此该条带可以通过冗余算法恢复出丢失的块,从而避免丢失该条带的数据。由于通过这种方案,即使存储设备没有备份电源,也能避免数据丢失,保证数据存储的可靠性,因此存储设备无需安装备份电源,因此可以将备份电源在存储设备占用的内部空间节省出来,让存储设备可以腾出更多内部空间放置存储介质,从而提高存储设备的存储容量。并且,可以节省制造备份电源的成本,从而节约存储设备的成本,此外,可以避免备份电源对存储设备的寿命造成限制。

[0011] 可选地,所述一批次写入的块中属于同一条带的块的数量小于或等于N个,包括:当所述N等于1,所述一批次写入的块中每个块均属于不同条带。

[0012] 通过这种可选方式,考虑到了校验块是1个还是多个的情况。如果条带中有1个校验块,对于S个条带中的任一个条带而言,由于一批次写入的块中每个块均属于不同条带,执行任意批次写入后,M+N个存储设备同时或近似同时缓存的块中,属于该条带的块也最多只有1个,那么M+N个存储设备在任一时间点同时掉电后,M+N个存储设备缓存的块中属于该条带的块最多只有1个,因此M+N个存储设备丢失的该条带的块最多只有1个,因此该条带可以通过剩余的M+N-1个块恢复出来。如果条带中有多个校验块,对于S个条带中的任一个条带而言,由于每批次写入的块中属于同一条带的块的数量小于或等于N个,将任意批次的块写入后,M+N个存储设备同时或近似同时缓存的块中,属于该条带的块的数量为1个至N个,那么M+N个存储设备在任一时间点同时掉电后,M+N个存储设备缓存的块中属于该条带的块最多只有N个,因此M+N个存储设备丢失的该条带的块最多只有N个,因此该条带可以通过剩余的块恢复出来。

[0013] 可选地,对于所述M+N个存储设备中的同一个存储设备,所述一批次写入的块中向所述存储设备写入的块与上一批次写入的块中向所述存储设备写入的块,属于不同的条带。

[0014] 通过这种可选方式,由于对于同一存储设备而言,一批次写入的块中向该存储设备写入的块与上一批次写入的块中向该存储设备写入的块属于不同的条带,那么如果上一

批次向存储设备*i*写入了条带*j*的块,则本批次会将条带*j*之外的其他条带的块写入存储设备*i*,而将条带*j*的块写入存储设备*i*之外的其他存储设备,因此可以保证同一条带的不同块分别在不同的存储设备存储,比如7+1结构的条带的8个块可以分别在8个不同的SSD存储,而对于EC算法或RAID算法等冗余算法而言,通常要求条带的不同块存储在不同的存储设备,才能够进行数据恢复,因此这种方式可以满足冗余算法的要求。

[0015] 可选地,所述*S*根据所述*M*、所述*N*以及*K*获取,所述*K*为每个存储设备最多缓存的块的数量。

[0016] 可选地,所述*S*通过以下公式获取:

$$[0017] \quad S \geq \text{ceil} \left( \frac{(M+N) * K}{N} \right)$$

[0018] 其中,ceil表示向上取整运算。

[0019] 通过这种可选方式,将*S*个条带组成一组,在每次写入过程中,可以将同一组中的不同条带一起写入,从而提高了这一组条带整体的存储效率。

[0020] 可选地,所述将所述一批次写入的块分别写入*M+N*个存储设备中之后,所述方法还包括:如果检测到掉电事件,获取所述*S*个条带中的第一块以及第二块,所述第一块为尚未写入至所述*M+N*个存储设备的块,所述第二块为所述*M+N*个存储设备已持久化存储的块;根据所述第一块以及所述第二块,对所述*S*个条带中丢失的块进行数据恢复。

[0021] 通过这种可选方式,如果存储设备由于掉电而丢失缓存的块,可以找到尚未写入到存储设备的块以及已经在存储设备上持久化存储的块,利用这些没有丢失的剩余的块,将丢失的块恢复出来,从而避免条带丢失,提高存储的可靠性。

[0022] 第二方面,提供了一种数据存储装置,所述装置具有实现上述第一方面或第一方面任一种可选方式中数据存储的功能。该装置包括至少一个模块,所述至少一个模块用于实现上述第一方面或第一方面任一种可选方式所提供的数据存储方法。

[0023] 第三方面,提供了一种计算设备,所述计算设备包括处理器以及存储器,所述存储器中存储有至少一条指令,所述指令由所述处理器执行以实现第一方面或第一方面任一种可选方式所述的方法。

[0024] 第四方面,提供了一种存储系统,所述存储系统包括如上述第三方面或第三方面任一种可选方式所述的计算设备以及*M+N*个存储设备。

[0025] 第五方面,提供了一种计算机可读存储介质,所述存储介质中存储有至少一条指令,所述指令由处理器执行以实现第一方面或第一方面任一种可选方式所述的方法。

[0026] 第六方面,提供了一种计算机程序,所述计算机程序用于执行上述第一方面或第一方面任一种可选方式所述的方法。

[0027] 第七方面,提供了一种芯片,所述芯片包括处理电路和与所述处理电路内部连接通信的输出接口,所述处理电路用于执行以实现第一方面或第一方面任一种可选方式所述的方法,所述输出接口用于将所述一批次写入的块分别写入*M+N*个存储设备中。

## 附图说明

[0028] 图1是本申请实施例提供的一种数据存储方法的实施环境的架构图;

[0029] 图2是本申请实施例提供的一种计算设备的结构示意图;

- [0030] 图3是本申请实施例提供的一种存储设备的结构示意图；
- [0031] 图4是本申请实施例提供的一种SSD的结构示意图；
- [0032] 图5是本申请实施例提供的一种存储系统的架构图；
- [0033] 图6是本申请实施例提供的一种数据存储方法的流程图；
- [0034] 图7是本申请实施例提供的一种条带和块的示意图；
- [0035] 图8是本申请实施例提供的一种条带和块的示意图；
- [0036] 图9是本申请实施例提供的一种获取条带的流程图；
- [0037] 图10是本申请实施例提供的一种写入顺序的示意图；
- [0038] 图11是本申请实施例提供的一种写入顺序的示意图；
- [0039] 图12是本申请实施例提供的一种掉电后恢复丢失的块的示意图；
- [0040] 图13是本申请实施例提供的一种数据恢复的流程图；
- [0041] 图14是本申请实施例提供的一种数据存储装置的结构示意图。

### 具体实施方式

[0042] 为使本申请的目的、技术方案和优点更加清楚,下面将结合附图对本申请实施方式作进一步地详细描述。

[0043] 以下,对本申请涉及的术语进行示例性介绍。

[0044] 独立磁盘冗余阵列(redundant arrays of independent drives,RAID):一种数据保护方法,RAID将多个物理的磁盘驱动器组合成一个或多个逻辑单元,从而能够对数据进行冗余存储。

[0045] 纠删码(erasure code,EC):一种数据保护方法,EC会将数据划分为数据块,使用冗余算法,对数据块进行扩展和编码,存储在一组不同的位置或存储介质中。

[0046] 条带(stripe,也称数据条带或data stripe):在RAID或EC技术中,对逻辑上的顺序数据进行分段以后,形成的一段连续数据。通常来讲,条带会被进一步划分为数据块,以便按照RAID或EC算法,对数据块计算出校验块,并将不同的块写入不同的存储设备。

[0047] 块(chunk,也称为strides、stripe unit或条带单元):在RAID或EC技术中,写入至存储设备或从存储设备读取的一段顺序数据,块一般由条带划分而来。

[0048] 固态硬盘(solid state drives,SSD,也称为固态设备、固态盘或闪存盘):一种主要以NAND flash memory(NAND闪存,NAND表示not AND,非与)集成电路组件作为持久性存储器的存储设备。

[0049] 以下,示例性介绍本申请提供的实施环境。

[0050] 图1是本申请实施例提供的一种数据传输方法的实施环境的架构图,该实施环境包括计算设备101、存储设备102以及客户端103。

[0051] 示例性地,计算设备101可以是服务器或者存储阵列控制器等。存储设备102可以是SSD、机械硬盘(hard disk drive,HDD)等。客户端103可以是任何数据访问设备,如应用服务器、主机或终端等。

[0052] 计算设备101以及存储设备102可以为客户端103提供数据存储服务。具体来说,客户端103可以向计算设备101提供待存储的数据,计算设备101可以从客户端103获取数据,将数据存储至存储设备102。示例性地,客户端103可以向计算设备101发送输入/输出

(input/output, I/O) 请求, 该I/O请求携带需要存储的数据, 计算设备101可以从客户端103接收I/O请求, 从I/O请求中获取数据, 存储数据。

[0053] 在一些可能的实施例中, 存储设备102可以作为云存储服务提供给用户。具体来说, 存储设备102可以运行在云环境中, 例如可以运行在公有云、私有云或混合云上。用户可以使用终端103, 在云存储服务中申请一定容量的存储空间。计算设备101可以为用户分配相应容量的存储空间, 例如为用户分配一个或多个存储设备102, 从而在分配的存储空间中, 存储数据。作为示例, 存储设备102可以提供为对象存储服务(英文: object storage service, 简称: OBS)、云硬盘、云数据库等。

[0054] 本领域技术人员可以知晓, 上述实施环境中的计算设备101、存储设备102以及客户端103的数量可以更多或更少。比如, 计算设备101可以仅为一个, 或者计算设备101为几十个或几百个或者更多数量, 此时上述实施环境还包括其他计算设备101。

[0055] 其中, 每个计算设备101连接的存储设备102的数量可以大于或等于M+N, M表示数据块的数量, N表示校验块的数量, 比如, 如果一个条带包括7个数据块以及1个校验块, 存储设备102的数量可以大于或等于8个。可选地, 多个计算设备101以及多个存储设备102可以组成存储集群, 通过协同运行来共同提供存储服务。

[0056] 以下, 示例性介绍本申请提供的硬件结构。

[0057] 本申请提供了一种计算设备, 该计算设备可以是上述实施环境中的计算设备101。

[0058] 如图2所示, 计算设备包括一个或一个以上处理器201以及一个或多个存储器202。

[0059] 处理器201可以是一个或多个中央处理器(central processing unit, CPU)、现场可编程门阵列(field-programmable gate array, FPGA)、专用集成电路(application specific integrated circuit, ASIC)可编程逻辑器件(programmable logic device, PLD)、控制器、状态机、门逻辑、分立硬件部件或者任何其它适合的电路。

[0060] 存储器202存储有至少一条指令, 处理器201用于执行该指令, 以实现下述方法实施例提供的步骤601、步骤602、步骤603、步骤604以及步骤605。

[0061] 存储器202可以是数据缓冲器(data buffer)。存储器202可以是易失性存储器, 例如动态随机存取存储器(dynamic random access memory, DRAM), 存储器也可以是非易失性存储器, 例如持久化内存(persistent memory)。

[0062] 在一些可能的实施例中, 该计算设备可以包括总线203, 处理器201以及存储器202之间可以通过总线203内部连接通信。

[0063] 在一些可能的实施例中, 该计算设备可以包括备份电源, 备份电源能够储存电能, 当计算设备运行时, 备份电源可以处于充电状态, 当计算设备掉电时, 备份电源可以提供电能, 从而对存储器202缓存的数据进行保护。比如如果存储器202是易失性存储器, 当计算设备掉电时, 备份电源可以供应电能, 可以通过备份电源提供的电能, 将存储器202缓存的数据存入其他存储空间, 从而避免存储器202缓存的数据由于掉电而丢失。其中, 备份电源可以是电池、电容或不间断电源(uninterruptible power system/uninterruptible power supply, UPS)。

[0064] 本申请提供了一种存储设备, 该存储设备可以是上述实施环境中的存储设备102。

[0065] 如图3所示, 该存储设备可以包括一个或一个以上处理器301以及一个或多个缓存存储器(cache) 302以及一个或多个持久化存储器303。



[0066] 处理器301可以是一个或多个CPU、FPGA、ASIC、PLD、控制器、状态机、门逻辑、分立硬件部件或者任何其它适合的电路。

[0067] 缓存存储器302可以而不限于是DRAM。缓存存储器302是易失性存储器。如果存储设备掉电,比如如果存储设备所处的设备掉电、存储设备突然被拔出等情况下,缓存存储器302中的数据将会丢失。在下述方法实施例中,缓存存储器302可以用于对计算设备写入的块进行缓存。在一种可能的实现中,如果缓存存储器302最多缓存K个块,缓存存储器302可以当缓存有K个块后,将已缓存的块存入持久化存储器303,从而对块进行持久化存储。

[0068] 持久化存储器303可以是非易失性存储器,例如可以是闪存,具体可以是NAND flash memory。在下述方法实施例中,持久化存储器303可以用于从缓存存储器302接收块,对块进行持久化存储。

[0069] 在一些可能的实施例中,该存储设备还可以包括总线304,处理器301、缓存存储器以及持久化存储器303之间可以通过总线304内部连接通信。

[0070] 以存储设备为SSD为例,参见图4,图4为本实施例提供的一种SSD的结构示意图。该SSD可以包括SSD接口、SSD控制器、缓存存储器(cache)、NAND闪存。SSD接口可以与计算设备相连。如图4所示,如果计算设备向SSD发送块,块会从SSD接口输入至SSD,SSD控制器会将块缓存至缓存存储器,缓存存储器可以将块存入NAND闪存。另外,计算设备还可以通过SSD接口,向SSD提供电能。

[0071] 以下,示例性介绍本实施例提供的存储系统。

[0072] 本实施例提供了一种存储系统,该存储系统可以包括图2实施例所示的计算设备以及M+N个存储设备,计算设备可以和M+N个存储设备相连。例如,参见图5,如果M为7,N为1,计算设备可以和8个SSD连接,这8个SSD即图5中的SSD1至SSD8。

[0073] 其中,存储设备可以设置在计算设备的内部,例如,存储设备可以与计算设备的总线连接,计算设备可以通过总线与存储设备内部通信。当然,存储设备也可以设置在计算设备的外部,例如,计算设备可以通过交换设备或网络与存储设备102通信连接。以SSD作为存储设备为例,计算设备可以包括输入/输出(input/output,简称:IO)端口、芯片或适配器(adapter),通过交换设备或网络与存储设备连接。在一种可能的场景中,如果计算设备掉电,则计算设备连接的所有存储设备可能均会掉电,而通过执行下述实施例提供的方法,可以避免所有存储设备缓存的数据丢失。

[0074] 以下,示例性介绍本申请提供的方法流程。

[0075] 图6是本申请实施例提供的一种数据存储方法的流程图。该方法可以应用于计算设备,该方法可以包括下述步骤601至605:

[0076] 601、计算设备获取待存储的S个条带。

[0077] S表示条带的数量,S个条带中的每个条带包括(M+N)个块,(M+N)个块包括M个数据块以及N个校验块,M表示数据块的数量,N表示校验块的数量,S、M和N为正整数。N可以为1,也可以大于或等于2。例如,条带可以包括7个数据块以及1个校验块;又如,条带可以包括6个数据块以及2个校验块,再如,条带可以包括6个数据块以及3个校验块,本实施例对条带中校验块和数据块的具体数量不做限定。

[0078] 示意性地,参见图7和图8,图7示出了8个条带,每个条带包括7个数据块以及1个校验块,图8示出了12个条带,每个条带包括6个数据块以及2个校验块。其中,图7以及图8中的

每一行对应一个条带,图7以及图8中的中的每一列对应同一位置的块,图7以及图8中的每一个方格表示一个条带中的一个块。图7以及图8中, $D_{ij}$ 表示条带*i*位置*j*的数据块, $P_i$ 表示条带*i*的第一个校验块, $Q_i$ 表示条带*i*的第二个校验块,*i*为条带的标识,*i*为大于或等于0的正整数。*j*为条带中块的标识,*j*为大于或等于0的正整数。

[0079] 关于条带的获取过程,参见图9,可以包括下述步骤一至步骤三:

[0080] 步骤一、计算设备将数据缓存区的数据划分为*S*个条带。

[0081] 计算设备可以获取待存储的数据,将数据缓存在数据缓存区中。例如,计算设备可以通过I/O接口,从终端接收待存储的数据。

[0082] 步骤二、对于*S*个条带中的每个条带,计算设备将条带划分为*M*个数据块。

[0083] 步骤三、计算设备可以根据*M*个数据块,计算*N*个校验块。

[0084] 其中,计算设备可以采用冗余算法,对*M*个数据块进行编码,得到*N*个校验块。冗余算法可以而不限于是RAID或EC算法。

[0085] 在一种可能的实现中,计算设备可以根据*M*、*N*、*K*,确定*S*,其中,*K*为计算设备最多缓存的块的数量,*K*为正整数。*S*可以和*M*、*N*以及*K*正相关,*S*可以和*N*负相关。

[0086] 需要说明的一点是,*S*根据*M*、*N*以及*K*获取仅是可选方式,而非必选方式。在另一些可能的实施例中,也可以通过其他方式来确定*S*,本实施例并不对如何确定*S*的方式进行限定。

[0087] 在一种可能的实现中,计算设备可以通过以下公式,获取*S*;

$$[0088] \quad S \geq \text{ceil} \left( \frac{(M+N)*K}{N} \right);$$

[0089] 其中,ceil表示向上取整运算,例如,ceil(*x*)表示获取不小于*x*的最小整数。\*表示相乘,—表示相除。

[0090] 以下示例性描述*S*的这种计算方式的技术效果:

[0091] 以一个条带包括7个数据块以及1个校验块、一个存储设备最多能够缓存1个块为例,如果条带中丢失了1个块,则丢失了块后,基于冗余算法,能够通过剩余的块恢复出丢失的块;如果条带中丢失了2个或2个以上的块,则丢失了块后,无法通过剩余的块恢复出丢失的块。因此,一个条带中最多丢失1个块。在本实施例中,为了保证所有存储设备掉电时,同一条带所丢失的块不超过1块,每次写入过程中,同一条带中只有1个块会写入至存储设备。那么由于同一条带中不同块的写入至存储设备的时间点不同,同一条带中不同块的缓存至存储设备的时间点就也会不同,也即是,在任一个时间点,所有存储设备最多缓存同一条带的1个块,那么,即使在该时间点所有存储设备掉电,该条带也最多丢失1个块,可以通过剩余的7个块恢复出丢失的块。

[0092] 而通过这种方式,一个条带需要经过8次写入过程,该条带中的每个块才能均得到存储:在第一次写入过程中,对条带中第一个数据块进行写入,在第二次写入过程中,对条带中第二个数据块进行写入,在第三次写入过程中,对条带中第三个数据块进行写入,依次类推,在第八次写入过程中,对条带中第八个数据块进行写入。如果对每一个条带分别按照这种串行的方式,写入到各个存储设备,会导致存储效率较低。有鉴于此,可以通过上述计算方式,将*S*个条带组成一组,每一组共需要的条带的数量为 $(7+1)*1/1=8$ 个,通过将同一组中的8个条带中的不同块一起写入,从而提高了这一组条带整体的存储效率。例如参见图

10, 可以获取8个条带, 每次分别写入这8个条带中的块。

[0093] 以一个条带包括6个数据块以及2个校验块、一个存储设备最多能够缓存3个块为例, 如果条带中丢失了1个或2个块, 则丢失了块后, 基于冗余算法, 能够通过剩余的块恢复出丢失的块; 如果条带中丢失了3个或3个以上的块, 则丢失了块后, 无法通过剩余的块恢复出丢失的块。因此, 一个条带中最多丢失2个块。在本实施例中, 为了保证所有存储设备掉电时, 同一条带所丢失的块不超过2块, 每3次写入过程中, 同一条带中最多有2个块会写入至存储设备。那么由于在相邻的3个时间点, 同一条带写入至所有存储设备的块的数量不超过2个, 在3个相邻的时间点, 同一条带缓存至所有存储设备的块的数量也就不超过2个, 也即是, 在任3个时间点, 所有存储设备最多缓存同一条带的2个块, 那么, 即使在该时间点所有存储设备掉电, 该条带也最多丢失2个块, 可以通过剩余的6个块恢复出丢失的块。

[0094] 而通过这种方式, 一个条带需要经过 $8*3/2=12$ 次写入过程, 该条带中的每个块才能均得到存储: 在第一次写入过程至第三次写入过程中, 对条带中第一个数据块以及第二个数据块进行写入, 在第四次写入过程至第六次写入过程中, 对条带中第三个数据块以及第四个数据块进行写入, 在第七次写入过程至第九次写入过程中, 对条带中第五个数据块以及第六个数据块进行写入, 在第十次写入过程至第十二次写入过程中, 对条带中第七个数据块至第八个数据块进行写入。如果对每一个条带分别按照这种串行的方式, 写入到各个存储设备, 会导致存储效率较低。有鉴于此, 为了避免影响存储效率, 可以将多个条带的块组成一组来进行存储, 每一组共需要的条带的数量为 $(6+2)*3/2=12$ 个。例如参见图8, 可以获取12个条带, 后续对这12个条带中的块依次写入。

[0095] 需要说明的一点是, 上述公式仅是S的计算方式的可选方式, 而非必选方式。在另一些可能的实施例中, 也可以通过上述公式之外的其他方式来计算S, 本实施例并不对S的计算方式进行限定。

[0096] 602、计算设备从S个条带确定一批次写入的块, 该一批次写入的块中属于同一条带的块的数量小于或等于N个。

[0097] 一批次写入的块包括向M+N个存储设备同时写入的块。具体地, 如果每个存储设备最多缓存1个块, 则一批次写入的块可以为M+N个块, 一批次写入的块中的每一个块对应一个存储设备, 在后续步骤603中, 会将一批次写入的块中的每一个块写入至对应的一个存储设备。其中, 同一批次写入的块对应同一个写入的时间点, 换句话说, 同一批次写入的块写入到M+N个存储设备的时间点会相同或近似相同。

[0098] 如果每个存储设备最多缓存K个块, K大于1, 则一批次写入的块可以为 $(M+N)*K$ 个块, 一批次写入的块中的K个块对应一个存储设备, 在后续步骤603中, 会将一批次写入的块中的K个块写入至该K个块对应的一个存储设备。其中, 同一批次写入的块对应K个写入的时间点, 具体地, 在第1个写入的时间点, 会将一批次写入的块中的M+N个块同时写入M+N个存储设备, 在第2个写入的时间点, 会将一批次写入的块中的另外M+N个块同时写入M+N个存储设备, 以此类推, 在第K个写入的时间点, 会将一批次写入的块中的最后M+N个块同时写入M+N个存储设备, 这时, 同一批次写入的块写入完成。

[0099] 为了便于表述, 在下文中, 将向M+N个存储设备同时写入的M+N个块, 定义为一次写入过程的块。如果每个存储设备最多缓存1个块, 则一批次写入的块为一次写入过程的块, 如果每个存储设备最多缓存K个块, 则一批次写入的块为连续K次写入过程的块。

[0100] 在一种可能的实现中,执行一次写入过程的实现方式可以为:计算设备运行 $M+N$ 个线程,每个线程将缓存的1个块写入至1个存储设备中, $M+N$ 个线程写入的时间点相同,则这种实现方式下,同一写入过程的 $M+N$ 个块写入的时间点相同。在另一种可能的实现中,执行一次写入过程的实现方式可以为:计算设备运行1个线程,该线程先将缓存的第1个块写入至存储设备,再将缓存的第2个块写入至下一个存储设备,依次类推,该线程将缓存的 $M+N$ 个块先后写入至对应的存储设备。其中,该线程将任一个块写入至存储设备后,可以无需等待该存储设备返回确认消息,而是直接将下一个块写入至下一个存储设备,这种实现方式下,同一次写入过程的 $M+N$ 个块写入的时间点近似相同。

[0101] 示例性地,如果 $K$ 为1,参见图10,第一批次写入的块可以为 $T_0$ 时间点向SSD1至SSD8同时写入的块,第二批次写入的块可以为时间点 $T_1$ 向SSD1至SSD8同时写入的块,第三批次写入的块可以为时间点 $T_2$ 向SSD1至SSD8同时写入的块,最后一批次写入的块可以为时间点 $T_7$ 向SSD1至SSD8同时写入的块。其中,向SSD1至SSD8同时写入的块为8个不同的块,示例性地,第一批次写入的8个块分别为向SSD1写入的块D00、向SSD2写入的块D10、向SSD3写入的块D20、向SSD4写入的块D30、向SSD5写入的块D40、向SSD6写入的块D50、向SSD7写入的块D60、向SSD8写入的块D70。

[0102] 示例性地,如果 $K$ 为3,参见图11,第一批次写入的块可以包括3次写入过程的块,第一批次中第一次写入过程的块为时间点 $T_0$ 向SSD1至SSD8同时写入的块,第一批次中第二次写入过程的块为时间点 $T_1$ 向SSD1至SSD8同时写入的块,第一批次中第三次写入过程的块为时间点 $T_2$ 向SSD1至SSD8同时写入的块。第二批次写入的块包括3次写入过程的块,第二批次中第一次写入过程的块为时间点 $T_3$ 向SSD1至SSD8同时写入的块,第二批次中第二次写入过程的块为时间点 $T_4$ 向SSD1至SSD8同时写入的块,第二批次中第三次写入过程的块为时间点 $T_5$ 向SSD1至SSD8同时写入的块。以此类推,最后一批次写入的块包括3次写入过程的块,最后一批次中第一次写入过程的块为时间点 $T_9$ 向SSD1至SSD8同时写入的块,最后一批次中第二次写入过程的块为时间点 $T_{10}$ 向SSD1至SSD8同时写入的块,最后一批次中第三次写入过程的块为时间点 $T_{11}$ 向SSD1至SSD8同时写入的块。

[0103] 本实施例中,由于每批次写入的块中属于同一条带的块的数量小于或等于 $N$ 个,对于 $S$ 个条带中的任一条带来说,执行任一批次中的任一写入过程后, $M+N$ 个存储设备中同时缓存的块中属于该条带的块的数量均小于或等于 $N$ 个。那么,如果 $M+N$ 个存储设备在任意时刻同时掉电,导致 $M+N$ 个存储设备缓存的数据全部丢失,由于 $M+N$ 个存储设备缓存的块中属于该条带的块的数量均小于或等于 $N$ 个,该条带最多也就丢失 $N$ 个块,因此条带即使丢失了块,可以通过剩余的块恢复,从而避免 $M+N$ 个存储设备同时掉电时丢失条带。

[0104] 具体来讲,计算设备执行任一次写入过程后, $M+N$ 个存储设备中的每个存储设备会接收一个块,缓存一个块,则 $M+N$ 个存储设备共计会缓存 $M+N$ 个块。其中,由于同一次写入过程的块写入的时间点相同或近似相同,那么 $M+N$ 个存储设备中每个存储设备接收到块的时间点会相同或近似相同,那么 $M+N$ 个存储设备中每个存储设备缓存块的时间点会相同或近似相同,因此会有 $M+N$ 个块同时缓存在 $M+N$ 个存储设备中。

[0105] 示例性地,如果 $K$ 为1, $N$ 为1,参见图10,每个时间点会有一行块缓存在8个SSD,这一行块中的每个块属于不同条带。计算设备在时间点 $T_0$ ,将第一批次的8个块:块D00、块D10、块D20至块D70,分别写入SSD1至SSD8,则SSD1至SSD8中的每个SSD会分别接收到块D00、块

D10、块D20至块D70这8个块中的一个块,分别缓存一个块,由于向8个SSD分别执行的8次写入的时间点相同或近似相同,那么8个SSD中每个SSD接收到块的时间点会相同或近似相同,那么8个SSD中每个SSD缓存块的时间点会相同或近似相同,因此,块D00、块D10、块D20至块D70的缓存时间点会相同或近似相同,由于块D00、块D10、块D20至块D70分别属于不同条带,因此即使8个SSD同时掉电,8个SSD缓存的数据全部丢失,导致块D00、块D10、块D20至块D70全部丢失,条带0、条带1至条带7这8个条带中的每个条带也只丢失了1个块,因此这8个条带中的每个条带均能够恢复,从而保证8个条带均不丢失。同理地,计算设备通过在时间点T1,将第二批次的8个块:块D71、块D01、块D11至块D61,分别写入SSD1至SSD8,则SSD1至SSD8中的每个SSD会分别接收到块D71、块D01、块D11至块D61这8个块中的一个块,分别缓存一个块,由于向8个SSD分别执行的8次写入的时间点相同或近似相同,因此块D71、块D01、块D11至块D61的缓存时间点会相同或近似相同,由于块D71、块D01、块D11至块D61分别属于不同条带,因此即使8个SSD同时掉电,8个SSD缓存的数据全部丢失,导致块D71、块D01、块D11至块D61全部丢失,条带0、条带1至条带7这8个条带中的每个条带也只丢失了1个块,因此这8个条带中的每个条带均能够恢复,从而保证8个条带均不丢失。

[0106] 示例性地,如果K为3,N为2,参见图10,每3个连续的时间点会有2行块缓存在8个SSD,这2行块中的同一条带的块的数量小于或等于2个。示意性地,计算设备在时间点T0至时间点T2,将第一批次的24个块:块D00、块D10、块D20、块D30、块D40、块D50、块D60、块D70、块D80、块D90、块DA0、块DB0、块D01、块D11、块D21、块D31、块D41、块D51、块D61、块D71、块D81、块D91、块DA1以及块DB1,分别写入SSD1至SSD8,其中第一批次的第一次过程会写入块D00、块D10至块D70,第一批次的第二次写入过程会写入块D80、块D90至块D31,第一批次的第三次写入过程会写入块D41、块D51至块DB1。由于在3个连续的时间点向8个SSD执行了3次写入过程,那么8个SSD中每个SSD会在3个连续的时间点接收到块,那么8个SSD中每个SSD会在3个连续的时间点缓存块,因此,块D00、块D10、块D20至块DB1这24个块会在3个连续的时间点缓存至8个SSD中。由于在块D00、块D10、块D20至块DB1中,每个条带最多存在2个块,因此即使8个SSD同时掉电,8个SSD缓存的数据全部丢失,导致块D00、块D10、块D20至块DB1全部丢失,条带0、条带1至条带7这8个条带中的每个条带也最多丢失了2个块,因此这8个条带中的每个条带均能够恢复,从而保证8个条带均不丢失。同理地,时间点T3至时间点T5写入第2批次的块后,也可以保证写入至持久化存储期间,8个条带不会因8个SSD同时掉电而丢失,时间点T6至时间点T8写入第3批次的块后,也可以保证8个条带不会因8个SSD同时掉电而丢失,时间点T9至时间点T12写入第3批次的块后,也可以保证8个条带不会因8个SSD同时掉电而丢失,综上所述,任一时间点写入任一批次的任一次写入过程的块后,8个条带均能够保证不丢失。

[0107] 在一个示例性场景中,参见图11,如果8个SSD中的每个SSD最多能缓存3个块,且条带包括6个数据块以及2个校验块,计算设备在时间点T0执行了第一次写入过程,在时间点T1执行了第二次写入过程、在时间点T2执行了第三次写入过程,以此类推,在时间点T11执行了第十二次写入过程。不妨设当前时间点为T7,则对于T7之前的3个时间点T4、T5以及T6来说,这3个时间点写入的块会在8个SSD中的每个SSD中缓存,而对于T4之前的时间点T0、T1、T2以及T3来说,这4个时间点写入的块已经被8个SSD中的每个SSD持久化存储。如果此时发生掉电,8个SSD会丢失缓存的时间点T4、T5以及T6的块,但对于任一条带而言,丢失的所

有块中该条带的块都没有超过2个,而由于6+2的EC算法可以容忍同一个条带丢失2个块,因此计算设备启动后,就可以使用EC算法,根据仍然被计算设备所保护的时间点T7至时间点T11的块,以及已经持久化存储至存储设备的时间点T0、T1、T2以及T3的块,恢复出时间点T4、T5以及T6的块。

[0108] 其中,如果N等于1,每批次写入的块中每个块均属于不同条带。例如,如果条带包括7个数据块以及1个校验块,每批次写入的8个块中的每个块均属于8个不同条带。比如,参见图10,第一批次写入的块包括块D00、块D10、块D20、块D30、块D40、块D50、块D60以及块D70,块D00属于条带0,块D10属于条带1,块D20属于条带2,块D30属于条带3,块D40属于条带4,块D50属于条带5,块D60属于条带6,块D70属于条带7。第二批次写入的块包括块D71、块D01、块D11、块D21、块D31、块D41、块D51以及块D61,块D71属于条带7,块D01属于条带0,块D11属于条带1,块D21属于条带2,块D31属于条带3,块D41属于条带4,块D51属于条带5,块D61属于条带6。

[0109] 此外,如果N大于1,每批次写入的块中属于同一条带的块的数量小于或等于N个。例如,如果条带包括6个数据块以及2个校验块,每批次写入对应的8个块中属于同一条带的块的数量可以小于或等于2个。即,每批次写入对应的8个块中的每个块可以均属于8个不同条带,或者,每批次写入对应的8个块中有2个块属于同一条带。又如,如果条带包括6个数据块以及3个校验块,每批次写入对应的9个块中属于同一条带的块的数量可以小于或等于3个。即,每批次写入对应的9个块中的每个块均属于9个不同条带,或者,每批次写入对应的9个块中有2个块属于同一条带,或者,每批次写入对应的9个块中有3个块属于同一条带。

[0110] 需要说明的一点是,步骤602是以一批次写入的块为例进行说明,同理地,计算设备可以确定多批次写入的块。其中,计算设备可以同时确定多批次写入的块,比如同时确定所有批次写入的块;计算设备也可以先确定一批次写入的块,将这批次写入的块写入后,再确定下一批次写入的块,本实施例对确定的时序不做限定。

[0111] 在一些可能的实施例中,对于M+N个存储设备中的同一个存储设备,一批次写入的块中向该存储设备写入的块与上一批次写入的块中向该存储设备写入的块,可以属于不同的条带。示例性地,参见图10,对于SSD1而言,第1批次写入的块中,向SSD1写入的块D00属于条带0,第2批次写入的块中,向SSD1写入的块D71属于条带7,第3批次写入的块中,向SSD1写入的块D62属于条带6,第4批次写入的块中,向SSD1写入的块D53属于条带5,第5批次写入的块中,向SSD1写入的块D44属于条带4,第6批次写入的块中,向SSD1写入的块D35属于条带3,第7批次写入的块中,向SSD1写入的块D26属于条带2,第8批次写入的块中,向SSD1写入的块P1属于条带8,通过这种顺序,8批次写入的块中,向SSD1写入的8个块分别属于8个不同的条带。示例性地,参见图11,对于SSD1而言,第1批次写入的块中,向SSD1写入的块D00属于条带0,第2批次写入的块中,向SSD1写入的块D72属于条带7,第3批次写入的块中,向SSD1写入的块D64属于条带6,第4批次写入的块中,向SSD1写入的块P5属于条带5,通过这种顺序,4批次写入的块中,向SSD1写入的8个块分别属于4个不同的条带。

[0112] 其中,如果一批次写入的块包括多次写入过程的块,同一批次中一次写入过程中向该存储设备写入的块与该批次中上一次写入过程中向该存储设备写入的块,可以属于不同的条带。例如,参见图11,第一批次的第二次写入过程中的块中,向SSD1写入的块D00属于条带8,第一批次的第三次写入过程的块中,向SSD1写入的块D00属于条带4,因此通过这种

顺序,第一批次的3次写入过程的块中,向SSD1写入的3个块也分别属于3个不同的条带。第二批次的第二次写入过程中的块中,向SSD1写入的块D33属于条带3,第二批次的第三次写入过程的块中,向SSD1写入的块DB3属于条带11,通过这种顺序,第二批次的3次写入过程的块中,向SSD1写入的3个块也分别属于3个不同的条带。

[0113] 通过这种方式,由于向同一存储设备写入的块属于不同的条带,那么同一条带的不同块就会分别在不同的存储设备存储,比如条带1的8个块可以分别在8个不同的SSD存储,而对于EC算法或RAID算法等冗余算法而言,通常要求条带的不同块存储在不同的存储设备,才能够进行数据恢复,因此这种方式可以满足冗余算法的要求。

[0114] 603、计算设备将一批次写入的块分别写入M+N个存储设备中。

[0115] 计算设备可以向M+N个存储设备分别发送一批次写入的块中的K个块,M+N个存储设备可以分别接收一批次写入的块中的K个块,分别缓存一批次写入的块中的K个块,分别对缓存的K个块进行持久化存储。其中,如果K大于1,对于任一个存储设备而言,计算设备可以按照时间的先后顺序,先将1个块写入至该存储设备,写入完成后,再将下1个块写入至该存储设备,直至将K个块写入至该存储设备。

[0116] 其中,如果存储设备最多缓存1个块,则存储设备每当接收到1个块时,存储设备可以对缓存的上一次写入的1个块进行持久化存储,将缓存中上一次写入的块更新为本次写入的1个块。比如,参见图10,如果SSD1最多缓存1个块,SSD1接收到D71时,可以对缓存中的D00进行持久化存储,将缓存中的D00更新为D71。如果存储设备最多缓存K个块,K大于1,则存储设备每当接收到1个块时,存储设备可以对缓存的一个或多个块进行持久化存储,将缓存中的块更新为本次写入的1个块。其中,可以优先持久化存储缓存时间点较远的比较老的块。例如,参见图10,如果SSD1最多缓存3个块,SSD1接收到D53时,SSD1的缓存包括D00、D71、D62,SSD1可以对缓存的D00进行持久化存储,删除缓存的D00,当然也可以对缓存的D00、D71均进行持久化存储,删除缓存的D00、D71,当然也可以对缓存的D00、D80、D41均进行持久化存储,删除缓存的D00、D80、D41。

[0117] 需要说明的是,步骤603是以一批次写入的块为例进行说明,同理地,计算设备可以将多批次写入的块分别写入M+N个存储设备中,直至将S个条带的所有块写入完成。具体地,计算设备可以按照时间的先后顺序,依次将每批次写入的块分别写入到M+N个存储设备中。相应地,M+N个存储设备可以按照时间的先后顺序,依次接收每批次写入的块,依次缓存每批次写入的块,对依次缓存的每批次写入的块进行持久化存储。例如,如果计算设备确定P个批次写入的块,计算设备会先将第一批次写入的块写入到M+N个存储设备中,再将第二批次写入的块写入到M+N个存储设备中,依次类推,直至将第P批次写入的块写入到M+N个存储设备中。相应地,M+N个存储设备会先接收到第一批次写入的块,缓存第一批次写入的块,再接收到第二批次写入的块,缓存第二批次写入的块,依次类推,直至接收到第P批次写入的块,缓存第P批次写入的块。

[0118] 示例性地,参见图10,计算设备可以通过确定8批次写入的块,将8批次写入的块写入8个SSD,来完成8个条带的存储。具体地,计算设备会将第一批次的8个块写入至8个SSD。即,将块D00写入至SDD1,将块D10写入至SDD2,将块D20写入至SDD3,将块D30写入至SDD4,将块D40写入至SDD5,将块D50写入至SDD6,将块D60写入至SDD7,将块D70写入至SDD8;再将第二批次的8个块写入至8个SSD,即,将块D71写入至SDD1,将块D01写入至SDD2,将块D11写入

至SDD3,将块D21写入至SDD4,将块D31写入至SDD5,将块D41写入至SDD6,将块D51写入至SDD7,将块D61写入至SDD8,以此类推,将第8批次的8个块写入至8个SSD,即,将块P1写入至SDD1,将块P2写入至SDD2,将块P3写入至SDD3,将块P4写入至SDD4,将块P5写入至SDD5,将块P6写入至SDD6,将块P7写入至SDD7,将块P0写入至SDD8,如此,条带0至条带7存储完成。

[0119] 需要说明的一点是,图10所示的写入顺序仅是一种可选实现方式,对于条带中包括7个数据块以及1个校验块,且存储设备最多缓存1个块的情况而言,在一些可能的实施例中,还可以通过图10之外的其他的写入顺序将8个条带写入到8个存储设备,只需保证每个条带的8个块写入到8个不同的存储设备,且每批次写入中每个条带写入到8个存储设备的块的总数量不超过1块,即可实现数据保护功能,能够在8个存储设备均掉电的情况下,避免8个条带中任一个条带被丢失。

[0120] 示例性地,参见图11,计算设备可以通过确定4批次写入的块,将4批次的块写入8个SSD,来完成12个条带的存储。具体地,计算设备会将第一批次的24个块,分为3次写入过程,写入至8个SSD。即,第一批次的第一次写入过程中,将块D00写入至SDD1,将块D10写入至SDD2,将块D20写入至SDD3,将块D30写入至SDD4,将块D40写入至SDD5,将块D50写入至SDD6,将块D60写入至SDD7,将块D70写入至SDD8;第一批次的第二次写入过程中,将块D80写入至SDD1,将块D90写入至SDD2,将块DA0写入至SDD3,将块DB0写入至SDD4,将块D01写入至SDD5,将块D11写入至SDD6,将块D21写入至SDD7,将块D31写入至SDD8,以此类推,第四批次的第三次写入过程中,将块Q9写入至SDD1,将块QA写入至SDD2,将块QB写入至SDD3,将块Q4写入至SDD4,将块Q5写入至SDD5,将块Q6写入至SDD6,将块Q7写入至SDD7,将块Q8写入至SDD8,如此,条带0至条带11存储完成。

[0121] 需要说明的一点是,图11所示的写入顺序仅是一种可选实现方式,对于条带中包括6个数据块以及2个校验块,且存储设备最多缓存3个块的情况而言,在一些可能的实施例中,还可以通过图11所示的写入顺序之外的其他的写入顺序,将12个条带写入到8个存储设备,只需保证每个条带的8个块写入到8个不同的存储设备,且每3次写入过程中每个条带写入到8个存储设备的块的总数量不超过2块,即可实现数据保护功能,能够在8个存储设备均掉电的情况下,避免12个条带中任一个条带被丢失。

[0122] 在一种可能的实现中,步骤601中获取到的每个条带的每个块可以通过数组中的元素表示,相应地,步骤602可以通过从数组中确定元素实现。

[0123] 具体来讲,S个条带可以表示为一个数组,该数组包括S行以及M+N列,S行中的每一行对应同一条带,例如第一行可以对应第一个条带,第二行可以对应第二个条带。M+N列中的每一列对应S个条带中的同一位置,例如第一列可以对应每个条带的第一个位置,第二列可以对应每个条带的第二个位置。数组的每个元素可以表示一个条带中的一个块,例如第i行第j个列的元素表示第i个条带的第j个块,i为小于或等于S的正整数,j为小于或等于M+N的正整数。示例性地,如果S个条带表示为数组dbuf[S][M+N],则dbuf[S][M+N]可以如下所示,其中,dbuf[S][M+N]中的每一个元素可以为图10或图11中的一个 $D_{ij}$ 。

[0124] dbuf[0][0]、dbuf[0][1]、……、dbuf[0][M+N-2]、dbuf[0][M+N-1]

[0125] dbuf[1][0]、dbuf[1][1]、……、dbuf[1][M+N-2]、dbuf[1][M+N-1]

[0126] dbuf[2][0]、dbuf[2][1]、……、dbuf[2][M+N-2]、dbuf[2][M+N-1]

[0127] ……



[0128]  $dbuf[s-2][0]$ 、 $dbuf[s-2][1]$ 、……、 $dbuf[s-2][M+N-2]$ 、 $dbuf[s-2][M+N-1]$

[0129]  $dbuf[s-1][0]$ 、 $dbuf[s-1][1]$ 、……、 $dbuf[s-1][M+N-2]$ 、 $dbuf[s-1][M+N-1]$

[0130] 在执行步骤602时,如果存储设备最多缓存1个块,可以从数组的第一列开始,遍历数组的每一列,每当确定M+N个元素,则将确定的M+N个元素对应的块,作为一批次写入的块。其中,如果S等于M+N,第M+N行的元素恰好为数组中最后一行的元素。那么,可以确定第一列的元素对应的块,作为第一批次写入的块,确定第二列的元素对应的块,作为第二批次写入的块,以此类推,确定第M+N列的元素对应的块,作为第M+N批次写入的块。如果S大于M+N,数组中第M+N行的元素之后还有其他行的元素。那么,可以确定第一行第一列的元素至第M+N行第一列的元素对应的块,作为第一批次写入的块;从M+N+1行第一列的元素开始,确定下一行的元素,如果确定的元素已经达到M+N个,则将已确定的M+N个元素对应的块,作为第二批次写入的块,如果确定到第S行第一列的元素,而已确定的元素的数量仍未达到M+N个,则继续从第一行第二列的元素开始,确定下一行的元素,直至确定了M+N个元素,将确定的M+N个元素对应的块,作为第二批次写入的块,依次类推。此外,如果存储设备能够缓存K个块,且K大于1,确定块的方式与此同理,相区别的是,每当确定 $(M+N)*K$ 个元素,则将确定的 $(M+N)*K$ 个元素对应的块,作为一批次写入的块。

[0131] 其中,为了避免将同一条带的多个块写入同一存储设备,在从数组中确定元素的过程中,可以将 $(M+N)*K$ 个元素作为一次循环,每当确定了 $(M+N)*K$ 个元素时,可以将上一次循环的第一个块待写入的存储设备,作为本次循环的第一个元素待写入的存储设备。也即是,如果上一次确定出 $(M+N)*K$ 个元素时,将第i个存储设备,作为上一次确定的第一个元素对应的块待写入的存储设备,则本次确定出 $(M+N)*K$ 个元素时,可以将第i+1个存储设备,作为本次确定的第一个元素对应的块待写入的存储设备。而对于本次循环的第二个元素至最后一个元素中的任一个元素而言,可以将上一个元素对应的块待写入的存储设备的下一个存储设备,作为当前元素对应的块待写入的存储设备。

[0132] 例如,参见图10,如果M为7,N为1,K为1,每8个元素可以为1次循环,第一次循环对应的元素为 $dbuf[0][0]$ 、 $dbuf[1][0]$ 至 $dbuf[7][0]$ , $dbuf[0][0]$ 待写入的SSD为SSD1,则 $dbuf[1][0]$ 之后开始,上一个元素待写入第i个SSD,则下一个元素待写入第i+1个SSD,则 $dbuf[1][0]$ 待写入的SSD为SSD2, $dbuf[2][0]$ 待写入的SSD为SSD3,以此类推, $dbuf[7][0]$ 待写入的SSD为SSD8。

[0133] 第二次循环对应的元素为 $dbuf[0][1]$ 、 $dbuf[1][1]$ 至 $dbuf[7][1]$ ,由于 $dbuf[0][1]$ 和 $dbuf[0][0]$ 都属于条带0,为了保证条带0的不同块写入到不同SSD,要避免 $dbuf[0][1]$ 和 $dbuf[0][0]$ 一样写入至SSD1;并且,由于 $dbuf[1][1]$ 和 $dbuf[1][0]$ 都属于条带1,为了保证条带1的不同块写入到不同SSD,要避免 $dbuf[1][1]$ 和 $dbuf[1][0]$ 一样写入至SSD2,所以,可以将SSD2作为 $dbuf[0][1]$ 待写入的SSD,则从 $dbuf[1][1]$ 开始,上一个元素待写入第i个SSD,则下一个元素待写入第i+1个SSD, $dbuf[1][1]$ 待写入的SSD为SSD3, $dbuf[2][1]$ 待写入的SSD为SSD4,以此类推, $dbuf[6][1]$ 待写入的SSD为SSD8,而从 $dbuf[7][1]$ 开始,重新将第1个SSD作为待写入的SSD,则 $dbuf[7][1]$ 待写入的SSD为SSD1。

[0134] 第三次循环对应的元素为 $dbuf[0][2]$ 、 $dbuf[1][2]$ 至 $dbuf[7][2]$ ,由于 $dbuf[0][2]$ 和 $dbuf[0][1]$ 都属于条带0,为了保证条带0的不同块写入到不同SSD,要避免 $dbuf[0][2]$ 和 $dbuf[0][1]$ 一样写入至SSD2;并且,由于 $dbuf[1][2]$ 和 $dbuf[1][1]$ 都属于条带1,为了

保证条带1的不同块写入到不同SSD,要避免dbuf[1][2]和dbuf[1][1]一样写入至SSD3,所以,可以将SSD3作为dbuf[0][2]待写入的SSD,则从dbuf[1][2]开始,上一个元素待写入第i个SSD,则下一个元素待写入第i+1个SSD,则dbuf[1][2]待写入的SSD为SSD4,dbuf[2][2]待写入的SSD为SSD5,以此类推,dbuf[5][2]待写入的SSD为SSD8,而从dbuf[5][2]开始,重新将第1个SSD作为待写入的SSD,则dbuf[6][2]待写入的SSD为SSD1,dbuf[7][2]待写入的SSD为SSD2。

[0135] 例如,参见图11,如果M为6,N为2,K为3,每24个元素可以为1次循环,第一次循环对应的元素为dbuf[0][0]、dbuf[1][0]……dbuf[7][0]、dbuf[0][1]、dbuf[1][1]……dbuf[B][1],dbuf[0][0]待写入的SSD为SSD1,则从dbuf[1][0]开始,上一个元素待写入第i个SSD,则下一个元素待写入第i+1个SSD,因此dbuf[1][0]待写入的SSD为SSD2,dbuf[2][0]待写入的SSD为SSD3,以此类推,dbuf[7][0]待写入的SSD为SSD8,而从dbuf[8][0]开始,重新将第1个SSD作为待写入的SSD,则dbuf[8][0]待写入的SSD为SSD1,dbuf[9][0]待写入的SSD为SSD2,以此类推,dbuf[3][1]待写入的SSD为SSD8,再从dbuf[4][1]开始,重新将第1个SSD作为待写入的SSD,则dbuf[9][0]待写入的SSD为SSD2,以此类推,dbuf[B][1]待写入的SSD为SSD8。

[0136] 第二次循环对应的元素为dbuf[0][2]、dbuf[1][2]……dbuf[B][2]、dbuf[0][3]、dbuf[1][3]……至dbuf[B][3],由于dbuf[0][2]和dbuf[0][0]都属于条带0,为了保证条带0的不同块写入到不同SSD,要避免dbuf[0][2]和dbuf[0][0]一样写入至第一个SSD;并且,由于dbuf[1][2]和dbuf[1][0]都属于条带1,为了保证条带1的不同块写入到不同SSD,要避免dbuf[1][2]和dbuf[1][0]一样写入至第二个SSD,所以,可以将SSD2作为dbuf[0][2]待写入的SSD,从dbuf[1][2]开始,上一个元素待写入第i个SSD,则下一个元素待写入第i+1个SSD,则dbuf[1][2]待写入的SSD为SSD3,dbuf[2][2]待写入的SSD为SSD4,以此类推,dbuf[6][2]待写入的SSD为SSD8,dbuf[7][2]待写入的SSD为SSD1。而对于dbuf[7][2]的下一个元素dbuf[8][2]而言,同理地,由于SSD2是第二次循环中的第一个待写入的盘,因此和dbuf[0][2]一样,将SSD2作为dbuf[8][2]待写入的SSD,那么,dbuf[9][2]待写入的SSD为SSD3,从dbuf[9][2]开始,上一个元素待写入第i个SSD,则下一个元素待写入第i+1个SSD,因此dbuf[A][2]待写入的SSD为SSD4,dbuf[B][2]待写入的SSD为SSD5,以此类推,dbuf[2][3]待写入的SSD为SSD8,而从dbuf[3][3]开始,重新将第1个SSD作为待写入的SSD,则dbuf[3][3]待写入的SSD为SSD1。而对于dbuf[3][3]的下一个元素dbuf[4][3]而言,同理地,由于SSD2是第二次循环中的第一个待写入的盘,因此和dbuf[0][2]一样,将SSD2作为dbuf[4][3]待写入的SSD,以此类推。

[0137] 第三次循环对应的元素为dbuf[0][4]、dbuf[1][4]……dbuf[B][4]、dbuf[0][5]、dbuf[1][5]……至dbuf[B][5],由于dbuf[0][4]和dbuf[0][2]都属于条带0,为了保证条带0的不同块写入到不同SSD,要避免dbuf[0][4]和dbuf[0][2]一样写入至SSD2;并且,由于dbuf[1][4]和dbuf[1][2]都属于条带1,为了保证条带1的不同块写入到不同SSD,要避免dbuf[1][4]和dbuf[1][2]一样写入至SSD3,所以,可以将SSD3作为dbuf[0][4]待写入的SSD,那么,从dbuf[1][4]开始,上一个元素待写入第i个SSD,则下一个元素待写入第i+1个SSD,则dbuf[1][4]待写入的SSD为SSD4,dbuf[2][4]待写入的SSD为SSD5,以此类推。

[0138] 其中,如果数组为上面所示的dbuf[s][M+N],列号表示为y,在确定元素的过程中,

如果当前元素的列号 $y*S/(M+N)$ 的余数为0,可以确定当前元素为本次循环的第一个元素,如果当前元素的列号 $y*S/(M+N)$ 的余数不为0,可以确定当前元素不为本次循环的第一个元素。因此,在确定元素的过程中,对于 $dbuf[0][y]$ ,在碰到 $y*s/(m+n)$ 的余数为0的 $dbuf[0][y]$ 元素之前,可以按照第一个存储设备至第 $M+N$ 个存储设备的顺序,将每个存储设备作为对应元素对应的块待写入的存储设备。例如,如果条带包括6个数据块以及2个校验块,每个存储设备能够缓存3个块,则 $y*S/(M+N)$ 的余数为0的 $y$ 值分别为0、2、4、6,则如果将第一个存储设备作为 $dbuf[0][0]$ 待写入的存储设备,在遇到 $dbuf[0][2]$ 之前,可以将下一个存储设备作为下一个元素待写入的存储设备。在遇到 $dbuf[0][2]$ 时,可以将第二个存储设备作为 $dbuf[0][2]$ 待写入的存储设备,之后,在遇到 $dbuf[0][4]$ 前,继续将下一个存储设备作为下一个元素待写入的存储设备。在遇到 $dbuf[0][4]$ 时,可以将第三个存储设备作为 $dbuf[0][4]$ 待写入的存储设备,之后继续将下一个存储设备作为下一个元素待写入的存储设备,依次类推。

[0139] 通过这种排序方式,在执行步骤603时,如果本次写入过程是从第 $x$ 个存储设备开始写入的,则会依次写入第 $x+1$ 个存储设备、第 $x+2$ 个存储设备,直至写入第 $M+N$ 个存储设备,继续从第一个存储设备开始写,直到写入第 $x-1$ 个存储设备。然后,执行下一次写入过程,继续从第 $x$ 个存储设备开始写入,依次写入第 $x+1$ 个存储设备、第 $x+2$ 个存储设备,直至写入第 $M+N$ 个存储设备,继续从第一个存储设备开始写,直到写入第 $x-1$ 个存储设备。如果进入下一轮循环,即写入了 $(M+N)*K$ 个块,则从第 $x+1$ 个存储设备开始写入,则会依次写入第 $x+2$ 个存储设备、第 $x+3$ 个存储设备,直至写入第 $M+N$ 个存储设备,继续从第一个存储设备开始写,直到写入第 $x$ 个存储设备。

[0140] 示例性地,如果数组为上面所示的 $dbuf[s][M+N]$ , $S=M+N$ ,则写入顺序可以如下所示:

[0141]  $dbuf[0][0]$ 、 $dbuf[1][0]$ 、 $dbuf[2][0]$ 、……、 $dbuf[s-2][0]$ 、 $dbuf[s-1][0]$

[0142]  $dbuf[s-1][1]$ 、 $dbuf[0][1]$ 、 $dbuf[1][1]$ 、……、 $dbuf[s-3][1]$ 、 $dbuf[s-2][1]$

[0143] ……

[0144]  $dbuf[2][s-2]$ 、 $dbuf[3][s-2]$ 、……、 $dbuf[0][s-2]$ 、 $dbuf[1][s-2]$

[0145]  $dbuf[1][s-1]$ 、 $dbuf[2][s-1]$ 、……、 $dbuf[s-2][s-1]$ 、 $dbuf[0][s-1]$

[0146] 其中,每一行表示一批次写入的块,每一列对应一个存储设备,每一个元素表示待在对应该批次写入向对应存储设备写入的块。例如, $dbuf[0][0]$ 、 $dbuf[1][0]$ 、……、 $dbuf[M+N-2][0]$ 、 $dbuf[s-1][0]$ 表示第一批次写入的块, $dbuf[0][0]$ 、 $dbuf[0][1]$ 、……、 $dbuf[0][M+N-1]$ 对应第一个存储设备, $dbuf[1][1]$ 表示第二批次写入向第一个存储设备写入的块。通过确定这样的写入顺序,在步骤603中,在将第一批次写入的块分别写入 $M+N$ 个存储设备时,就是按照第一行所示的顺序,将 $dbuf[0][0]$ 对应的块,即第一个条带的第一个块写入第一个存储设备,将 $dbuf[1][0]$ 对应的块,即第二个条带的第一个块写入第二个存储设备,以此类推,将 $dbuf[s-1][0]$ 对应的块,即第 $S$ 个条带的第一个块写入第 $M+N$ 个存储设备。在将第二批次写入的块分别写入 $M+N$ 个存储设备时,就是按照第二行所示的顺序,将 $dbuf[0][1]$ 对应的块,即第一个条带的第二个块写入第一个存储设备,将 $dbuf[1][1]$ 对应的块,即第二个条带的第二个块写入第二个存储设备,以此类推,将 $dbuf[s-2][1]$ 对应的块,即第 $S-1$ 个条带的第二个块写入第 $M+N$ 个存储设备,将 $dbuf[s-1][1]$ 对应的块,即第 $S$ 个条带的第二个块写

入第1个存储设备。

[0147] 需要说明的一点是,以上仅是对块的写入顺序的确定方式进行示例性描述,而非限定块的写入顺序的确定方式必须如此。在本实施例中,只需保证确定每批次写入的块之后,每批次写入的块中属于同一条带的块的数量小于或等于N个即可,本实施例对具体如何确定各个块的写入顺序不做具体限定。

[0148] 604、如果检测到掉电事件,计算设备获取S个条带中的第一块以及第二块。

[0149] 掉电事件即存储设备的电源停止向存储设备提供电能的事件,比如可以是存储设备被拔出、存储设备所在的设备掉电等事件,如果存储设备掉电,如果存储设备未安装备份电源,则存储设备缓存的数据可能发生丢失,则计算设备可以通过执行步骤604以及步骤605,恢复出丢失的数据。可选地,掉电事件可以是M+N个存储设备全部掉电的事件。在一个示例性场景中,计算设备可以是存储服务器,M+N个存储设备可以是插入至存储服务器的M+N个SSD,如果存储服务器掉电,则M+N个SSD均会掉电,存储服务器可以在重新上电后,执行本步骤。

[0150] 第一块为尚未写入至M+N个存储设备的块。具体来讲,第一块可以为尚未执行的每次写入过程的块。例如,如果S个条带共需通过W次写入过程写入完成,而掉电时已经执行了第一次写入过程至第P批次写入过程,第一块可以为第P+1次写入过程至第W次写入过程的块。例如,参见图12,12个条带通过12次写入过程写入完成,如果掉电时间点为时间点T6,而T6对应于第6次写入过程(即第3批第1次写入过程),那么掉电时计算设备已执行了第7次写入过程,尚未执行第8次写入过程至第12次写入过程,因此第一块可以为第8次写入过程至第12次写入过程的块,即时间点T7至时间点T11对应的块。

[0151] 关于获取第一块的具体过程,第一块可以预先缓存在计算设备的数据缓冲器(data buffer)中,则计算设备可以从数据缓冲器中,获取第一块。其中,如果数据缓冲器为易失性存储器,该数据缓冲器可以通过备份电源保护,当发生掉电时,可以通过备份电源提供的电能,将数据缓冲器中的第一块存入某一存储空间,当上电后,计算设备可以从该存储空间将第一块重新存入数据缓冲器,再从数据缓冲器读取第一块。如果数据缓冲器为非易失性存储器,在掉电期间,数据缓冲器中的第一块不会丢失,因此当上电后,计算设备可以直接从数据缓冲器读取第一块。

[0152] 第二块为M+N个存储设备已持久化存储的块。第二块可以不随着掉电而丢失。

[0153] 其中,如果S个条带共需通过W批次写入完成,每个存储设备最多缓存K次写入的块,而掉电时,计算设备已经执行了第一次写入过程至第P次写入过程,那么M+N个缓存设备缓存的块为第(P-K-1)次写入过程的块至第(P-1)批次写入过程的块,这些块可能尚未持久化存储,而在这些块之前被写入的块,即第一次写入过程至第(P-K)次过程的块已经被持久化存储,因此,第二块可以为第一次写入过程至第(P-K)次写入过程的块。

[0154] 例如,参见图12,12个条带共需通过12次写入过程写入完成,如果存储设备最多缓存3次写入的块,掉电时间点为时间点T6,则掉电时,M+N个存储设备缓存的块可以为时间点T4对应的块至时间点T6对应的块,即第5次写入过程的块至第7次过程写入的块,而时间点T4之前写入的块已经被持久化存储,因此第二块可以是时间点T0至时间点T3对应的块,即第1次写入过程至第4次写入过程的块。

[0155] 关于获取第二块的具体过程,第二块可以存储在存储设备的持久化存储介质中,

则计算设备可以从存储设备的持久化存储介质,读取第二块。以SSD为例,计算设备可以从SSD的NAND flash中,读取第二块。

[0156] 605、计算设备根据第一块以及第二块,对S个条带中丢失的块进行数据恢复。

[0157] 计算设备可以根据第一块以及第二块,检测S个条带中每个条带是否完整;如果任一条带中丢失了块,计算设备可以从第一块以及第二块中,获取该条带剩余的块;计算设备可以根据该条带剩余的块,通过RAID算法或者EC算法,恢复出该条带中丢失的块。其中,由于每批次写入的块中属于同一条带的块的数量小于或等于N个,因此在任意时间点掉电时,同一条带在M+N个存储设备缓存的块的数量会小于或等于N个,那么同一条带丢失的块的数量也会小于或等于N个,那么,每个条带即使丢失了块,也可以通过剩余的块以及冗余算法恢复出丢失的块,因此每个条带不会真正丢失,能够得到保护。

[0158] 示例性地,参见图13,数据恢复的流程可以包括以下步骤:

[0159] 步骤一、将掉电时数据缓冲器中的数据恢复出来。

[0160] 步骤二、读取存储设备的数据

[0161] 步骤三、根据数据缓冲器中的数据以及存储设备上读出的数据,检测数据是否完整,如果数据不完整,执行步骤四,如果数据完整,流程结束。

[0162] 步骤四、根据数据缓冲器中的数据以及存储设备上读出的数据,按照RAID算法或EC算法,恢复数据。

[0163] 步骤五、判断所有条带中丢失的数据是否均已恢复,如果存在任一条带的数据尚未恢复,继续执行步骤四,如果所有条带中的数据均已恢复,流程结束。

[0164] 本实施例提供了一种能够在所有存储设备同时掉电的情况下,不依赖于存储设备的备份电源,也能够避免存储设备缓存的条带丢失的方案。通过对各个条带中的各个块写入至存储设备的先后顺序进行改进,从将一批次写入同一条带的M+N个块,改进为一批次写入同一条带的小于或等于N个块,那么对于每个条带而言,由于将该条带中同时写入到M+N个存储设备的块的数量小于或等于N个,那么该条带中同时缓存至M+N个存储设备的块的数量也会小于或等于N个。因此,即使M+N个存储设备在任一时间点全部掉电,造成M+N个存储设备缓存的数据全部丢失,该条带中丢失的块的数量会小于或等于N个,因此该条带可以通过冗余算法恢复出丢失的块,从而避免丢失该条带的数据。由于通过这种方案,即使存储设备没有备份电源,也能避免数据丢失,保证数据存储的可靠性,因此存储设备无需安装备份电源,因此可以将备份电源在存储设备占用的内部空间节省出来,让存储设备可以腾出更多内部空间放置存储介质,从而提高存储设备的存储容量。并且,可以节省制造备份电源的成本,从而节约存储设备的成本,此外,可以避免备份电源对存储设备的寿命造成限制。

[0165] 图14是本申请实施例提供的一种数据存储装置的结构示意图,如图14所示,该装置包括:

[0166] 确定模块1401,用于执行步骤602;写入模块1402,用于执行步骤603。

[0167] 可选地,该一批次写入的块中属于同一条带的块的数量小于或等于N个,包括:当该N等于1,该一批次写入的块中每个块均属于不同条带。

[0168] 可选地,对于该M+N个存储设备中的每个存储设备,该一批次写入的块中向该存储设备写入的块与上一批次写入的块中向该存储设备写入的块,属于不同的条带。

[0169] 可选地,S根据M、N以及K获取,所述K为每个存储设备最多缓存的块的数量。

[0170] 可选地,该S通过以下公式获取:

$$[0171] \quad S \geq \text{ceil} \left( \frac{(M+N)*K}{N} \right)$$

[0172] 其中,ceil表示向上取整运算,该K为每个存储设备最多缓存的块的数量。

[0173] 可选地,该装置还包括:获取模块,用于执行步骤604;恢复模块,用于执行步骤605。

[0174] 需要说明的一点是,图14实施例提供的数据存储装置在存储数据时,仅以上述各功能模块的划分进行举例说明,实际应用中,可以根据需要而将上述功能分配由不同的功能模块完成,即将计算设备的内部结构划分成不同的功能模块,以完成以上描述的全部或者部分功能。另外,上述实施例提供的数据存储装置与数据存储方法实施例属于同一构思,其具体实现过程详见方法实施例,这里不再赘述。

[0175] 作为一种可能的产品形态,本申请实施例中的计算设备,可以由芯片来实现。该芯片包括处理电路和与该处理电路内部连接通信的输出接口;该处理电路用于执行步骤601、步骤602、步骤604以及步骤605;该输出接口用于执行步骤603。可选地,该芯片还可以包括存储介质,该存储介质用于存储处理电路执行的指令。

[0176] 上述所有可选技术方案,可以采用任意结合形成本申请的可选实施例,在此不再一一赘述。

[0177] 作为一种可能的产品形态,本申请实施例所述的计算设备,还可以使用下述来实现:一个或多个现场可编程门阵列(英文全称:field-programmable gate array,英文简称:FPGA)、可编程逻辑器件(英文全称:programmable logic device,英文简称:PLD)、控制器、状态机、门逻辑、分立硬件部件、任何其它适合的电路、或者能够执行本申请通篇所描述的各种功能的电路的任意组合。

[0178] 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中描述的各方法步骤和模块,能够以电子硬件、计算机软件或者二者的结合来实现,为了清楚地说明硬件和软件的可互换性,在上述说明中已经按照功能一般性地描述了各实施例的步骤及组成。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。本领域普通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申请的范围。

[0179] 所属领域的技术人员可以清楚地了解到,为了描述的方便和简洁,上述描述的系统、装置和模块的具体工作过程,可以参见前述方法实施例中的对应过程,在此不再赘述。

[0180] 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,所述模块的划分,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,例如多个模块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。另外,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、装置或模块的间接耦合或通信连接,也可以是电的,机械的或其它的形式连接。

[0181] 所述作为分离部件说明的模块可以是或者也可以不是物理上分开的,作为模块显示的部件可以是或者也可以不是物理模块,即可以位于一个地方,或者也可以分布到多个网络模块上。可以根据实际的需要选择其中的部分或者全部模块来实现本申请实施例方案

的目的。

[0182] 另外,在本申请各个实施例中的各功能模块可以集成在一个处理模块中,也可以是各个模块单独物理存在,也可以是两个或两个以上模块集成在一个模块中。上述集成的模块既可以采用硬件的形式实现,也可以采用软件功能模块的形式实现。

[0183] 所述集成的模块如果以软件功能模块的形式实现并作为独立的产品销售或使用时,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,本申请的技术方案本质上或者说对现有技术做出贡献的部分,或者该技术方案的全部或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,或者网络设备等)执行本申请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(read-only memory, ROM)、随机存取存储器(random access memory, RAM)、磁碟或者光盘等各种可以存储程序代码的介质。

[0184] 以上所述,仅为本申请的具体实施方式,但本申请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申请揭露的技术范围内,可轻易想到各种等效的修改或替换,这些修改或替换都应涵盖在本申请的保护范围之内。因此,本申请的保护范围应以权利要求的保护范围为准。

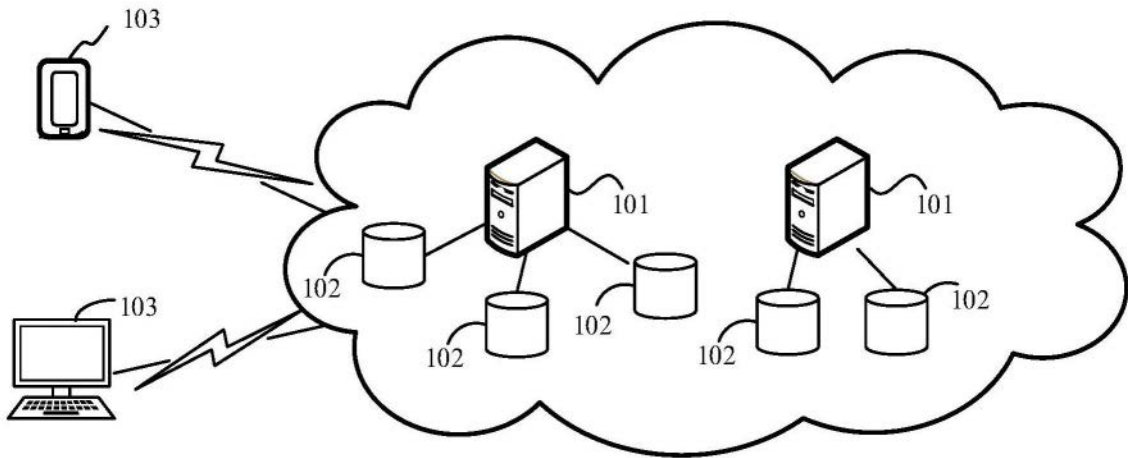


图1

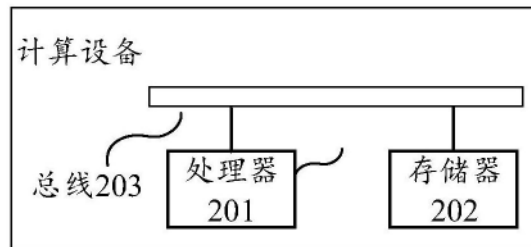


图2

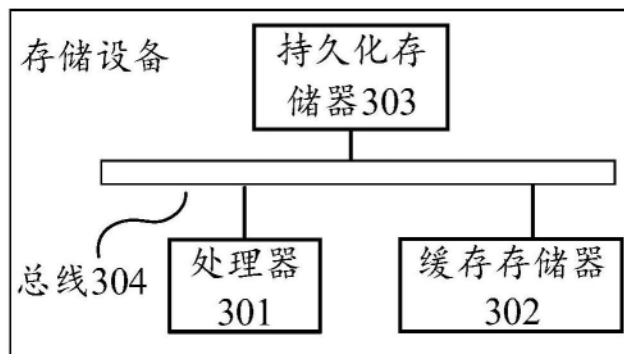


图3



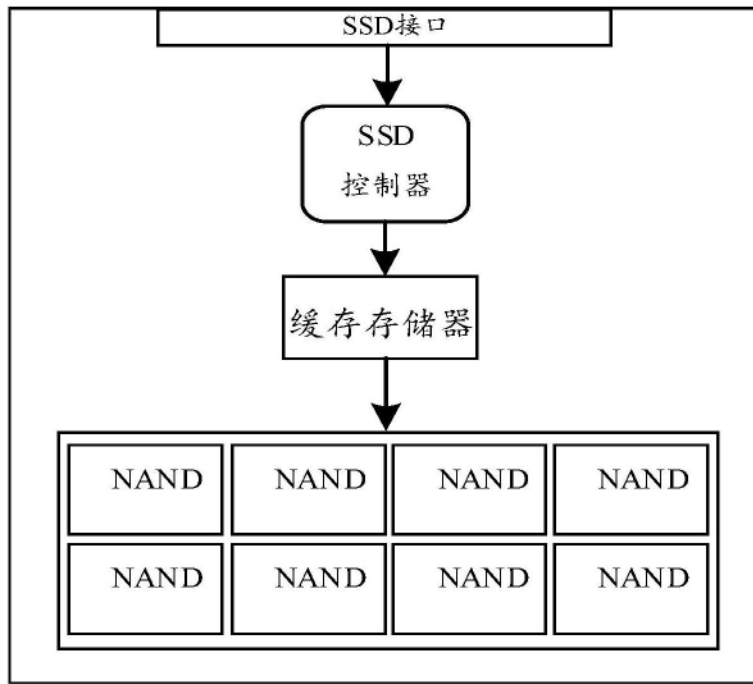


图4

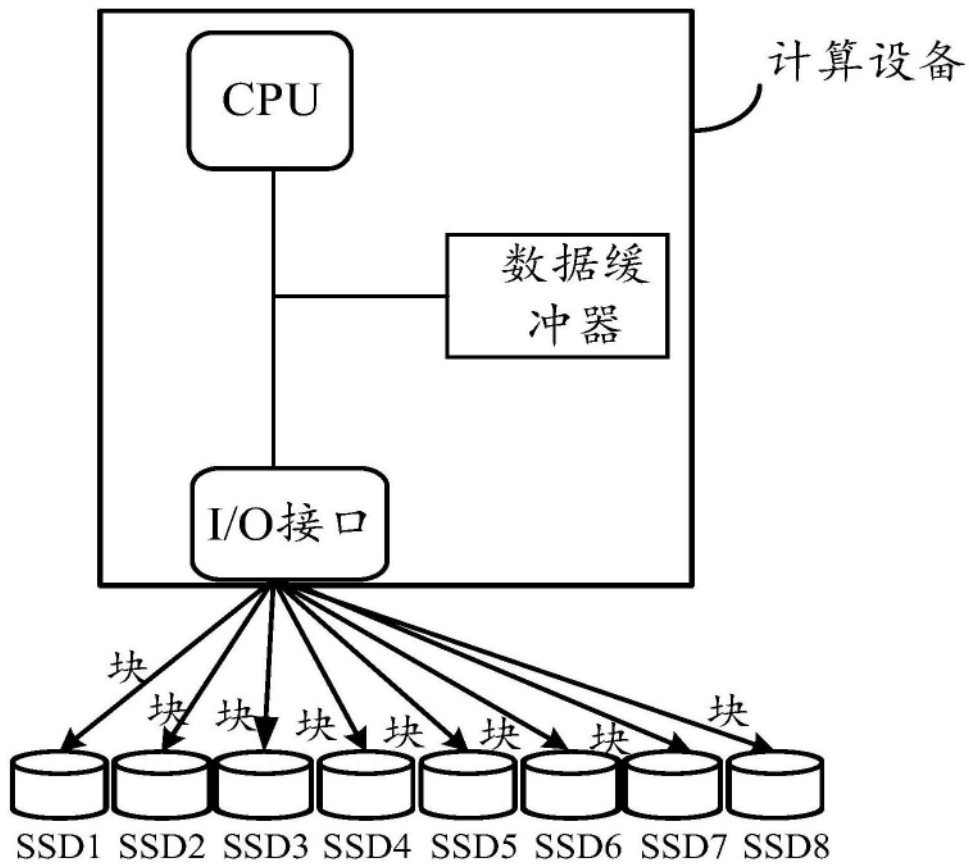


图5

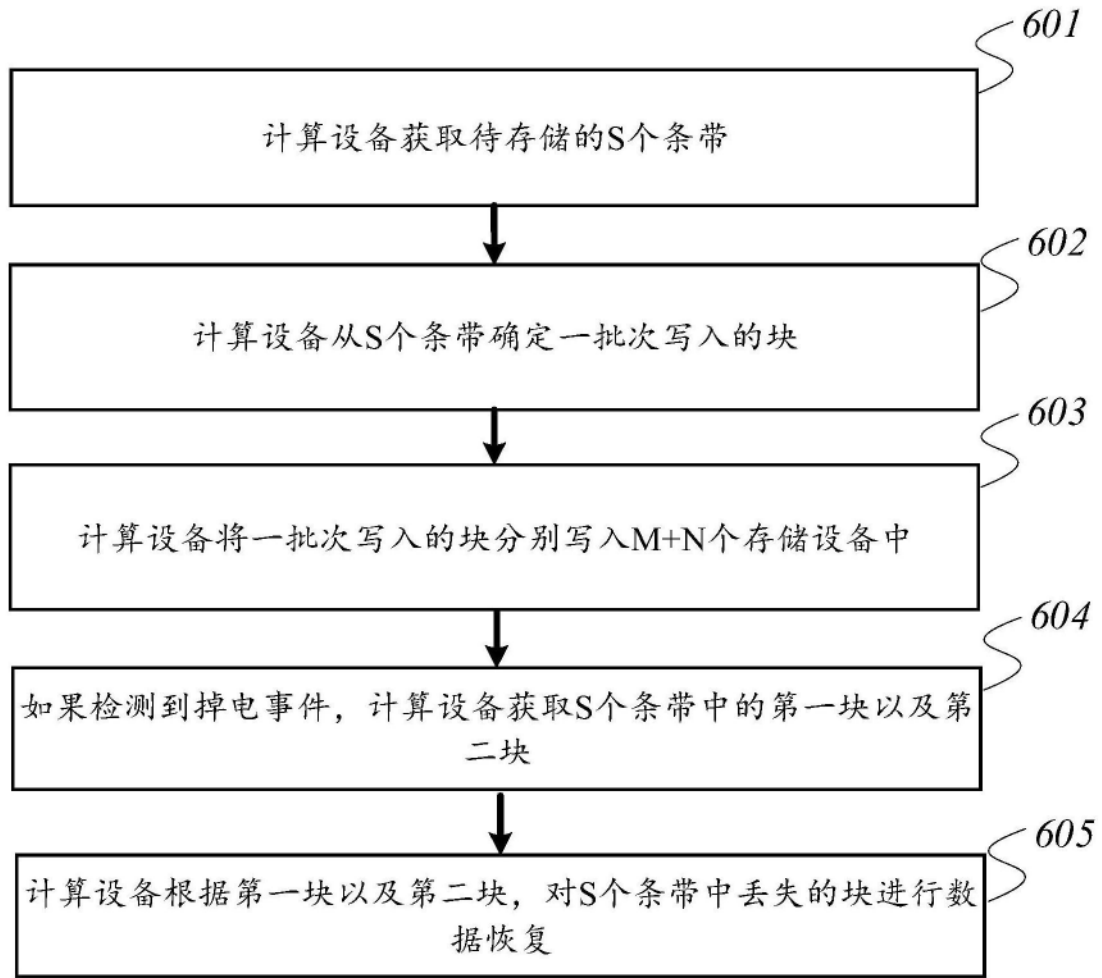


图6

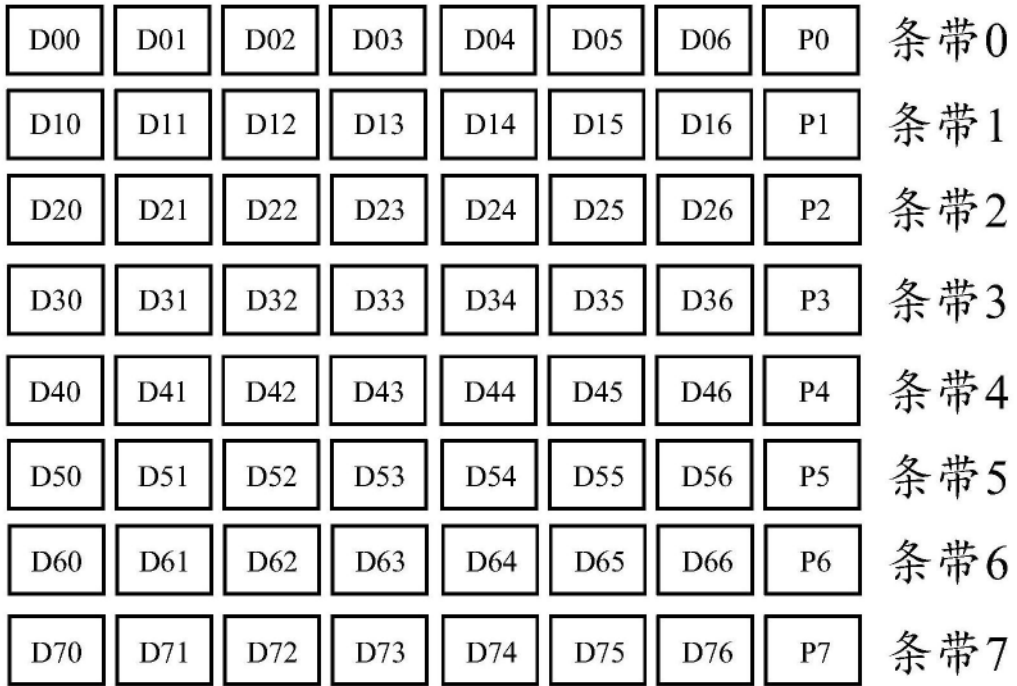


图7

D00	D01	D02	D03	D04	D05	P0	Q0	条带0
D10	D11	D12	D13	D14	D15	P1	Q1	条带1
D20	D21	D22	D23	D24	D25	P2	Q2	条带2
D30	D31	D32	D33	D34	D35	P3	Q3	条带3
D40	D41	D42	D43	D44	D45	P4	Q4	条带4
D50	D51	D52	D53	D54	D55	P5	Q5	条带5
D60	D61	D62	D63	D64	D65	P6	Q6	条带6
D70	D71	D72	D73	D74	D75	P7	Q7	条带7
D80	D81	D82	D83	D87	D85	P8	Q8	条带8
D90	D91	D92	D93	D94	D95	P9	Q9	条带9
DA0	DA1	DA2	DA3	DA4	DA5	PA	QA	条带10
DB0	DB1	DB2	DB3	DB4	DB5	PB	QB	条带11

图8

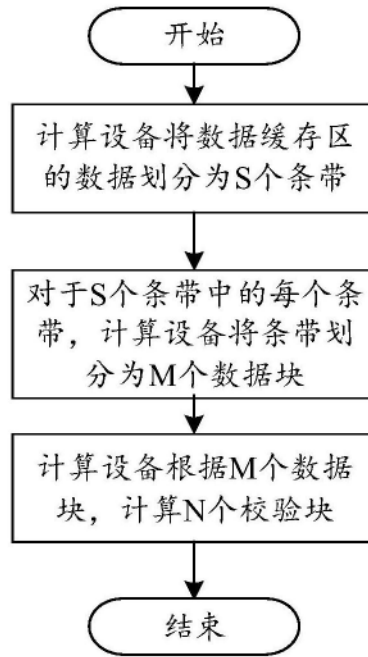


图9

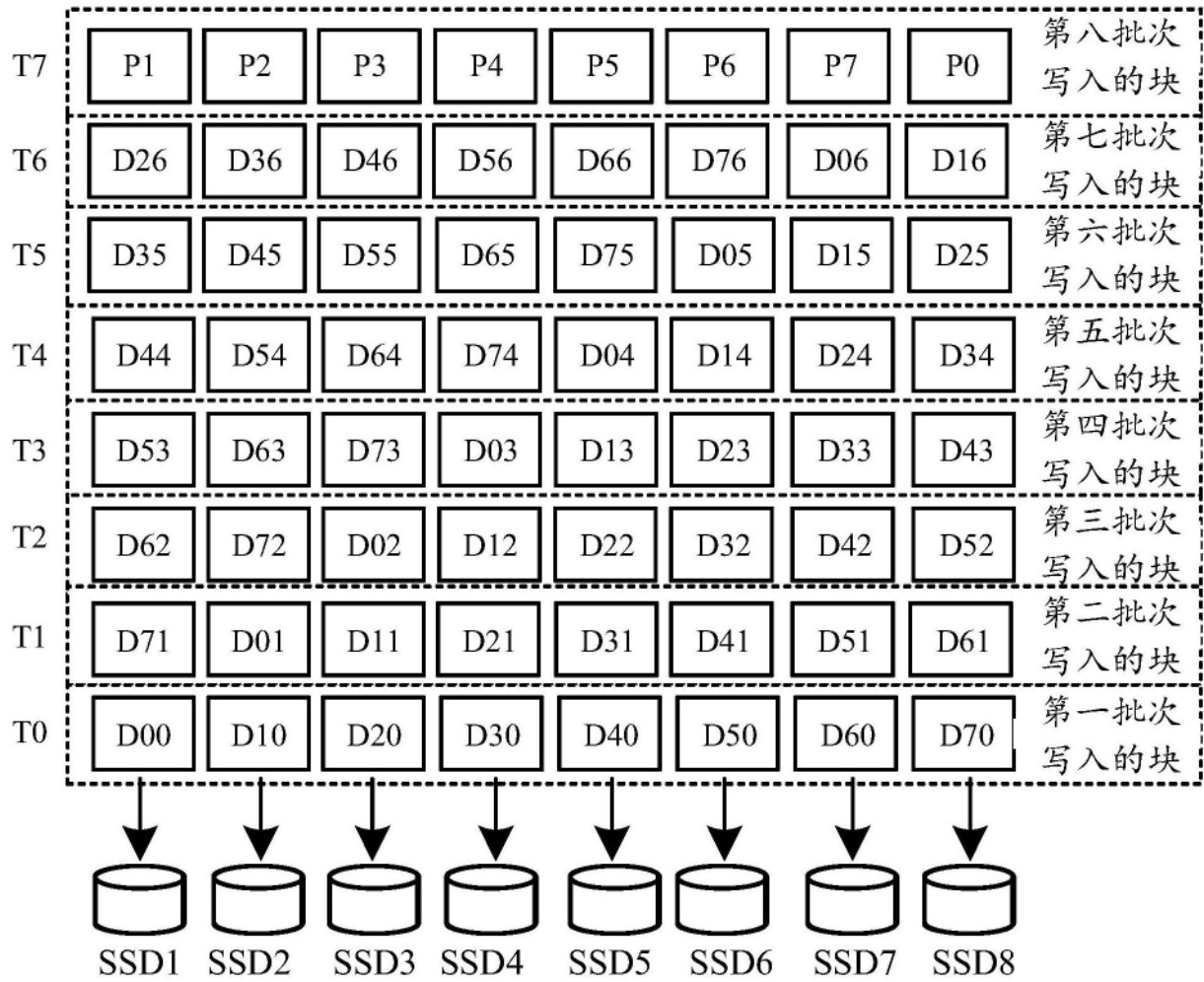


图10

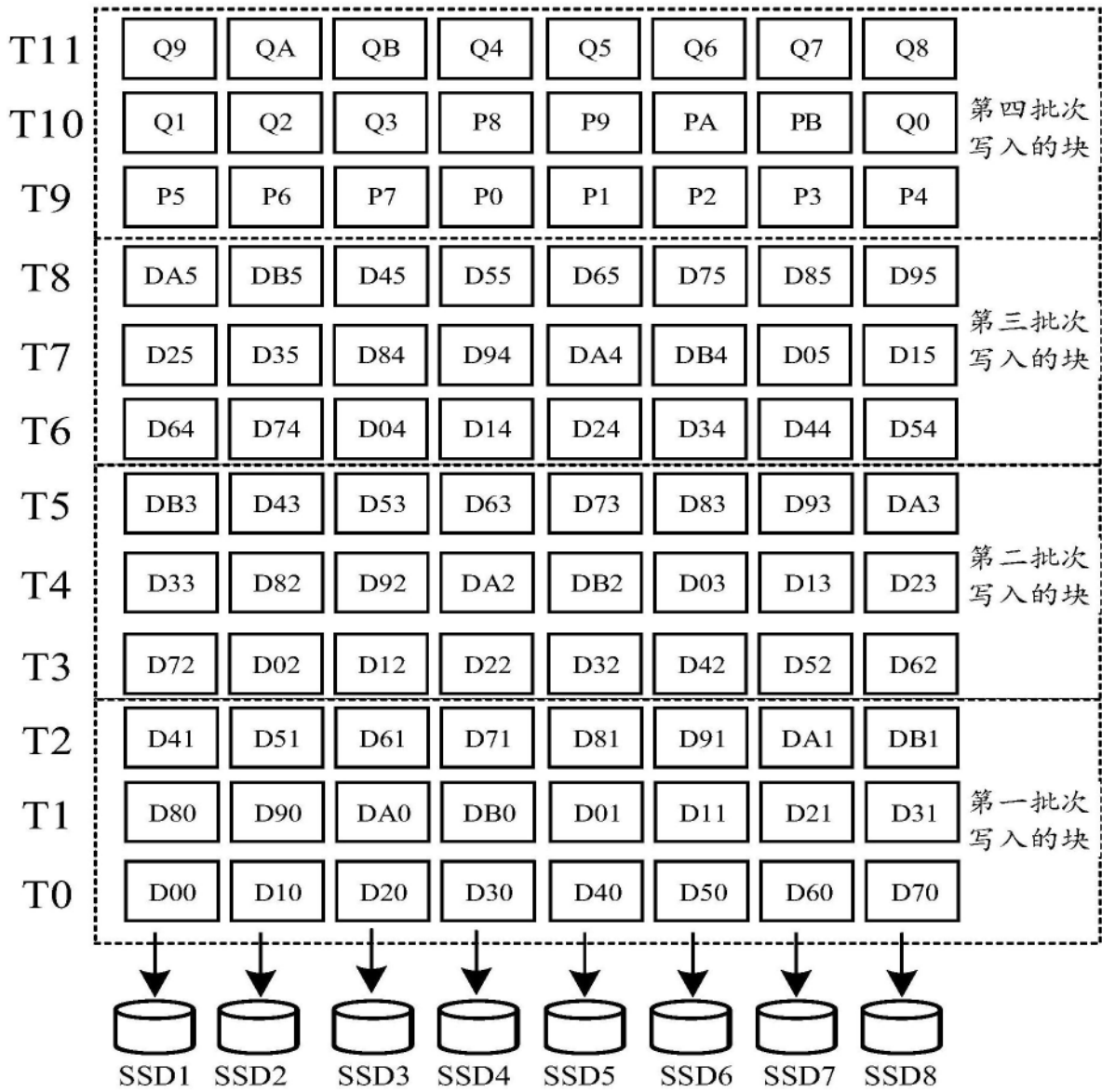


图11

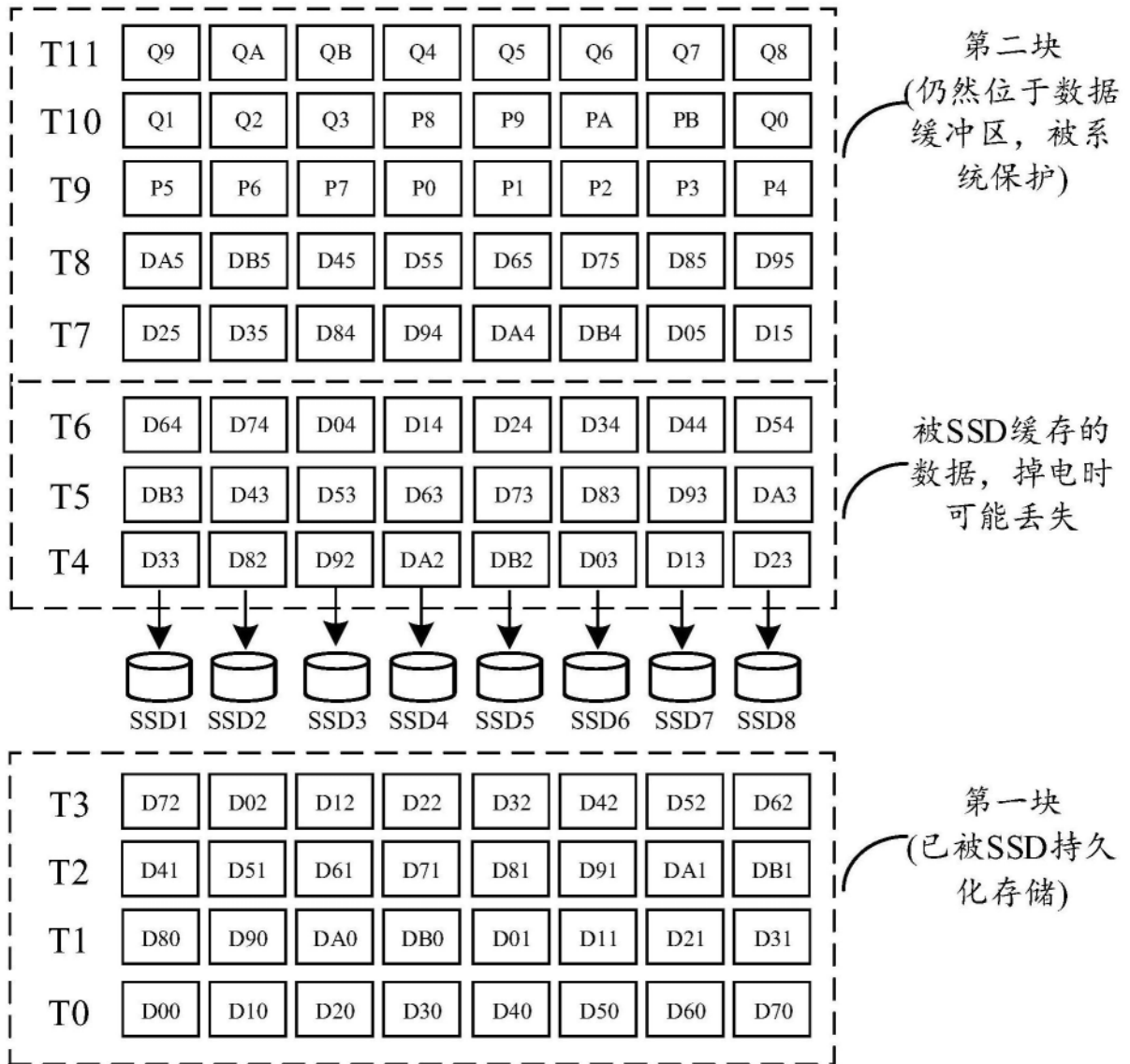


图12



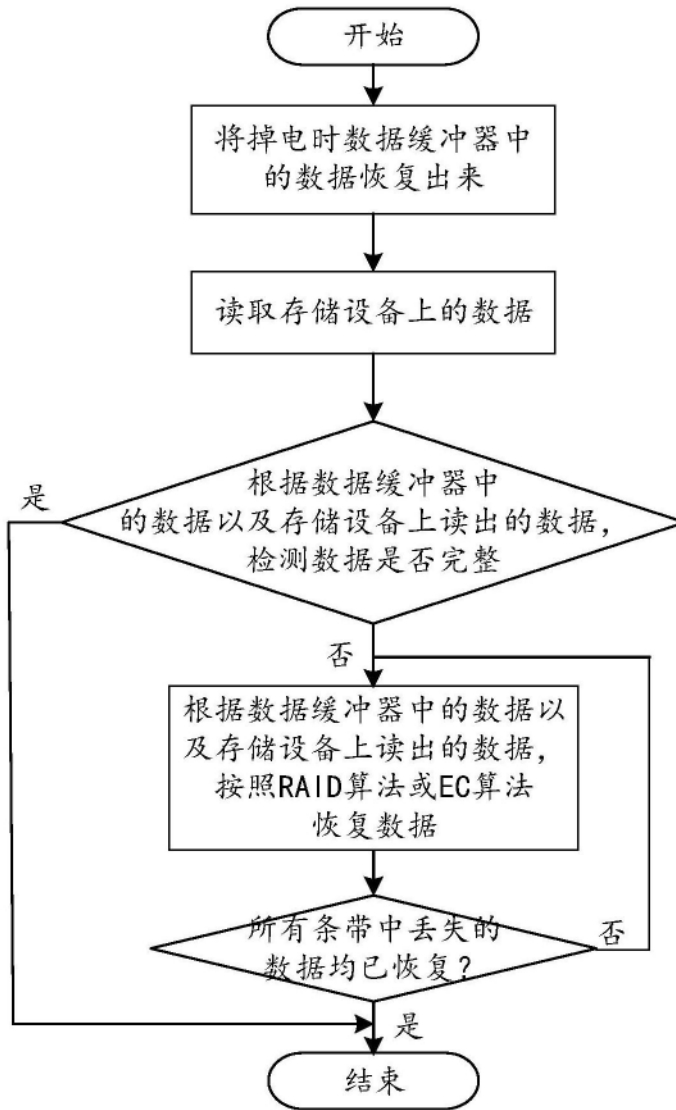


图13

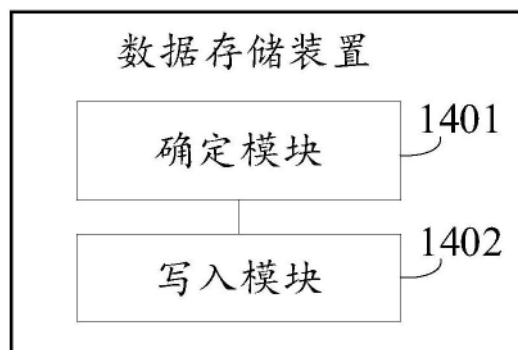


图14