

一种存储系统和系统垃圾回收方法

技术领域

本申请涉及存储技术领域，特别是一种存储系统和系统垃圾回收方法。

5 背景技术

闪存阵列是指包含固态硬盘和至少两个控制器的存储系统。其中，控制器以段（英文：segment）的形式管理固态硬盘的存储空间。段是由控制器分配的一段连续的逻辑地址的区域。段也是系统垃圾回收的基本单位。每个控制器都可以将接收的数据写入其管理的段中。固态硬盘从各个控制器接收
10 的数据按照接收的时间顺序连续写入固态硬盘的块（英文：block）中。因此，在一个时间段内，各个控制器发送的数据可能被写入同一个块中。因此，每个块中存储的数据可能对应不同控制器管理的段。

闪存阵列进行系统垃圾回收并释放一些段后，对于一个固态硬盘的一个块来说，这个块并没有被擦除干净，因为它仍然包含其他未进行系统垃圾回收
15 的段的数据。所以，当固态硬盘进行自己的垃圾回收时，仍然需要将这个块中其他未进行系统垃圾回收的段的数据写入空白的块中。导致固态硬盘内部出现的写放大的次数多。

发明内容

20 本申请提出了一种存储系统和系统垃圾回收方法，可以减少固态硬盘内部的写放大的次数。

本申请第一方面提供了一种存储系统。该存储系统包括第一控制器、第二控制器和固态硬盘。所述固态硬盘与所述第一控制器连接，并且所述固态硬盘与所述第二控制器连接。第一控制器或第二控制器以段为单位管理所述
25 固态硬盘的存储空间。第一控制器用于对所述第一控制器管理的段中的多个段进行系统垃圾回收。当第一控制器进行系统垃圾回收时，第二控制器对第

二控制器管理的段中的多个段进行系统垃圾回收。其中，第一控制器管理的段中的多个段是在一个时间段内由第一控制器分配的，第二控制器管理的段中的多个段也是在所述时间段内由第二控制器分配的。

在本申请提供的存储系统中，第一控制器对其管理的段中的多个段进行系统垃圾回收，第二控制器也对其管理的段中的多个段进行系统垃圾回收，并且第一控制器进行系统垃圾回收的段与第二控制器进行系统垃圾回收的段是在相同的时间段内分配的。由于段是第一控制器或第二控制器管理固态硬盘的存储空间单位，因此将数据写入段就是将数据写入固态硬盘的存储空间。第一控制器和第二控制器进行系统垃圾回收的段是在相同的时间段内分配的，因此这些段的数据写入固态硬盘的时间相近。进一步地，由于固态硬盘在写入数据时是按照接收数据的时间顺序连续写入一个或多个块，因此，相同的时间段内分配的段被执行系统垃圾回收之后，这些段所对应的块中的有效数据已经被全部复制到其他块中，这些段所对应的块中的数据就都变成无效数据了。当固态硬盘进行内部的垃圾回收时，被执行系统垃圾回收的段所对应的块无需再次进行有效数据的复制，从而减少了固态硬盘内部的写放大的次数。

结合第一方面，在第一方面的第一种实现中，第一控制器管理的段中的多个段是所述时间段内由第一控制器分配的所有的段。并且，第二控制器管理的段中的多个段是所述时间段内由第二控制器分配的所有的段。由此，可以保证在所述时间段内分配的所有的段都被系统垃圾回收了，那么固态硬盘中也有更多的块中的有效数据被全部复制到其他块，从而称为无效数据。

结合第一方面或者第一方面的第二种实现，在第一方面的第三种实现中，所述时间段在所述第一控制器或者第二控制器在所述时间段内分配的段的数量达到第一预设阈值时结束。按照这种实现方式，第一控制器和第二控制器便于统计各自分配的段的数量，操作方便。

结合第一方面或者第一方面的第二种实现，在第一方面的第四种实现

中,所述时间段在第一控制器以及第二控制器在所述时间段内分配的段的数量的总和达到第二预设阈值时结束。与第一方面的第三种实现方式相比,在第一方面的第四种实现方式中,第一控制器或者第二控制器需要给对方发送消息以查询对方分配的段的数量,操作起来较为复杂。然而,按照这种实现方式,在各个时间段内分配的段的数量上限相同。后续统计每个段组所包含的无效数据的数据量,从而选择出包含无效数据最多的段组时会更加准确。

结合第一方面或者第一方面的以上任意一种实现,在第一方面的第五种实现中,第一控制器管理的段中的多个段与第二控制器管理的段中的多个段属于同一个段组。这个段组包含的无效数据的数据量大于其他任意一个段组包含的无效数据的数据量。在第一方面的第五种实现中,第一控制器可以向第二控制器发送消息以查询第二控制器管理的段所包含的无效数据的数据量,第一控制器通过计算自己管理的段所包含的无效数据的数据量与第二控制器管理的段所包含的无效数据的数据量的和,从而得到包含无效数据最多的段组。这就保证了进行系统垃圾回收的段组是包含无效数据最多的段组,提高了系统垃圾回收的效率。

结合第一方面或者第一方面的第一种实现至第四种实现的任意一个,在第一方面的第六种实现中,第一控制器管理的段中的多个段与第二控制器管理的段中的多个段属于同一个段组。第一控制器管理的段中的多个段包含的无效数据的数据量大于第一控制器管理的段中的属于其他任意一个段组的多个段包含的无效数据的数据量。与第一方面的第五种实现相比,第一控制器只需统计自己管理的段所包含的无效数据的数据量,不需要向第二控制器发送消息以查询第二控制器管理的段所包含的无效数据的数据量。

结合第一方面的第五种实现或者第一方面的第六种实现,在第一方面的第七种实现中,第一控制器将所述段组的组标识发送给第二控制器。第二控制器根据预先保存的所述组标识与第二控制器管理的段中的多个段中每个段的标识之间的对应关系,确定第二控制器管理的段中的所述多个段。在第

七种实现中，由第一控制器确定系统垃圾回收的段组的组标识，将这个组标识发送给第二控制器，第二控制器可以根据所述组标识查找属于这个段组的多个段。

结合第一方面的第七种实现，在第一方面的第八种实现中，第一控制器管理的每个段的标识是第一控制器分配的，第一控制器管理的每个段的标识是第二控制器分配的。

结合第一方面的第七种实现，在第一方面的第九种实现中，第一控制器管理的每个段的标识是第一控制器分配的。第二控制器管理的每个段的标识也是第一控制器分配的，例如，第二控制器向第一控制器发送申请段标识的请求，以获得标识。

结合第一方面或者第一方面的以上任意一种实现，在第一方面的第十种实现中，所述第一控制器在执行系统垃圾回收时的具体操作是：将所述第一控制器管理的段中的多个段的每个段包含的有效数据的源逻辑地址发送给所述固态硬盘。第一控制器为所述有效数据分配目标逻辑地址并将所述目标逻辑地址发送给所述固态硬盘，以指示所述固态硬盘将所述有效数据从所述源逻辑地址复制到所述目标逻辑地址。然后，第一控制器向所述固态硬盘发送去映射命令。所述去映射命令包括所述第一控制器管理的段中的多个段的每个段的逻辑地址区间，以指示所述固态硬盘删除所述固态硬盘保存的所述每个段的逻辑地址与实际地址之间的对应关系。

本申请第二方面提供了一种系统垃圾回收方法。该方法应用于第一方面提供的存储系统中，由第一方面的第一控制器和第二控制器执行。

本申请第三方面提供了一种计算机程序产品，包括存储了程序代码的计算机可读存储介质，所述程序代码包括的指令可以由上述第一方面的第一控制器和第二控制器执行，并用于执行上述第二方面的至少一种方法。

附图说明

为了更清楚地说明本发明实施例的技术方案，下面将对实施例中所需要

使用的附图作简单地介绍。

图 1 是本发明实施例提供的应用场景图；

图 2 是本发明实施例提供的控制器的结构图；

图 3 是本发明实施例提供的写数据的方法流程图；

5 图 4 是本发明实施例提供的系统垃圾回收的方法流程图。

具体实施方式

本发明实施例提出了一种存储系统和垃圾回收方法，可以减少固态硬盘内部的写放大的次数。

图 1 描绘了本发明实施例提供的存储系统的组成图，图 1 所示的存储系
10 统包括至少两个控制器（如图所示的控制器 11 以及控制器 22）和多个固态硬盘 44。

控制器 11 和控制器 22 分别通过存储区域网络（英文：storage area network, SAN）与主机（图中未示出）连接。控制器 11 与控制器 22 之间可以通信。控制器 11 和控制器 22 具有相同的物理结构和功能。以控制器 11
15 为例，它可以是一种计算设备，如服务器、台式计算机等等。在控制器 11 上安装有操作系统以及应用程序。控制器 11 可以接收来自主机的输入输出（I/O）请求。控制器 11 还可以存储 I/O 请求中携带的数据（如果有的话），并且将该数据写入固态硬盘 44 中。

固态硬盘（英文：Solid State Disk, SSD）是以闪存（英文：flash memory）
20 芯片为存储介质的存储器，又名固态驱动器（Solid State Drive, SSD）。

图 1 仅是示例性说明，在实际应用中存储系统可包含多个控制器，每个控制器的物理结构和功能与控制器 11 类似，并且本实施例并不限定控制器 11 与控制器 22 之间，以及任意一个控制器与固态硬盘 44 之间的连接方式。只要各个控制器之间，以及各个控制器和固态硬盘 44 之间能够相互通信。

25 图 2 是控制器 11 的结构示例图，如图 2 所示，控制器 11 包括接口卡 110、

处理器 112 和接口卡 113。

接口卡 110 用于和主机通信，控制器 11 可以通过接口卡 110 接收主机的操作指令。处理器 112 可能是一个中央处理器(英文:central processing unit, CPU)。在本发明实施例中，处理器 112 可以用于接收来自主机的 I/O 请求、
5 处理所述 I/O 请求。所述 I/O 请求可以是写数据请求或者读数据请求，处理器 112 还可以将写数据请求中的数据发送给固态硬盘 44。此外，处理器 112 还可以用于执行系统垃圾回收操作。

可选地，控制器 11 还可以包括存储器 111。存储器 111 用于临时存储从主机接收的数据或从固态硬盘 44 读取的数据。控制器 11 接收主机发送的多
10 个写数据请求时，可以将所述多个写数据请求中的数据暂时保存在存储器 111 中。当存储器 111 的容量达到一定阈值时，将存储器 111 存储的数据以及为所述数据分配的逻辑地址发送给固态硬盘 44。固态硬盘 44 存储所述数据。存储器 111 包括易失性存储器，非易失性存储器或其组合。易失性存储器例如为随机访问存储器(英文: random-access memory, RAM)。非易失性
15 存储器例如软盘、硬盘、固态硬盘(solid state disk, SSD)、光盘等各种可以存储程序代码的机器可读介质。存储器 111 具有保电功能，保电功能是指系统发生掉电又重新上电时，存储器 111 中存储的数据也不会丢失。

接口卡 113，用于和固态硬盘 44 通信，控制器 11 可以通过接口卡 113 将写数据请求(包括数据以及控制器 11 为所述数据分配的逻辑地址)发送
20 给固态硬盘 44 存储。

控制器 22 的结构，以及各个部件的功能与控制器 11 类似，这里不再赘述。另外，控制器 11 和控制器 22 都属于系统控制器，系统控制器是独立的设备，不同于固态硬盘中的控制芯片。本实施例将固态硬盘的控制芯片称为固态硬盘控制器。如无特别指明，本实施例中的控制器都是指系统控制器。

固态硬盘 44 包括固态硬盘控制器和存储介质。其中，固态硬盘控制器用于执行控制器 11 或控制器 22 发送的写数据请求或者读数据请求等操作。

固态硬盘控制器中包含闪存翻译层（英文：flash translation layer, FTL）。闪存翻译层用于保存有数据的逻辑地址与实际地址之间的对应关系。因此，闪存翻译层用于将控制器发送的写数据请求或者读数据请求中的逻辑地址转化为固态硬盘中数据的实际地址。数据的逻辑地址是由控制器分配的，一个段的逻辑地址区间的子集。数据的逻辑地址包括起始逻辑地址和长度，起始逻辑地址指示所述数据位于的所述段的位置，长度代表所述数据的大小。数据的实际地址可以是固态硬盘中该数据的物理地址，也可以是在所述物理地址的基础上经过虚拟化，只对固态硬盘控制器可见的地址。该经过虚拟化的实际地址对系统控制器不可见。

10 存储介质通常包括一个或多个闪存芯片。每个闪存芯片包括若干个块。闪存芯片在读取或写入时是以一个页（英文：page）为基础的，但擦除操作只能以一个块为基础，擦除操作是指将这个块的所有位都设置为“1”。在擦除之前，固态硬盘控制器需要先将这个块中的有效数据复制到另一个块的空白页中去。块中的有效数据是指该块中保存的没有被修改过的数据，这部分数据可能会被读取。块中的无效数据是指该块中保存的已经被修改过的数据，这部分数据不可能被读取。

每个块包含多个页（英文：page）。固态硬盘在执行写数据请求时，是以页为单位来写数据的。举例来说，控制器 11 向固态硬盘控制器发送一个写数据请求。所述写数据请求包括数据的逻辑地址。固态硬盘控制器在接收
20 所述写数据请求之后，按照接收的时间顺序将所述数据连续写入一个或多个块中。连续写入一个或多个块是指，固态硬盘查找一个空白的块，将数据写入所述空白的块，直至将所述空白的块填满，当所述数据的大小超过块的容量时，固态硬盘控制器再查找下一个空白的块，继续写入。闪存翻译层建立并保存所述逻辑地址与写入所述数据的页的实际地址之间的对应关系。当控
25 制器 11 向固态硬盘控制器发送读数据请求，要求读取所述数据时，所述读数据请求中包括所述逻辑地址。固态硬盘控制器根据所述逻辑地址、以及所

述逻辑地址与实际地址之间的对应关系读取所述数据，并将该数据发送给控制器 11。

固态硬盘中的块、页以及实际地址对控制器来说是不可见的，因此控制器不能直接访问固态硬盘中的块和页。为了进行系统垃圾回收，系统控制器

5 将固态硬盘的存储空间按照设定大小划分为若干个段，段是系统垃圾回收的基本单位，每个段均是多个逻辑地址的集合，并且这些逻辑地址是连续的。例如，所述设定大小是 1024KB。一个段对应的逻辑地址区间是 0KB-1023KB；另一个段对应的逻辑地址区间是 1024KB-2047 KB；再一个段对应的逻辑地址区间是 2048 KB -3071 KB，依此类推。

10 在将数据写入段时，可以以扇区为单位写入。例如，一个扇区包括 520 个字节。其中 512 个字节是数据，8 个字节是 512 字节的数据的数据完整性字段（英文：data integrity field，DIF）。也就是说，每次写入的数据须是 512 个字节的整数倍。如果所述数据的大小不满足 512 个字节的整数倍，可以添

15 据写入段之后，需要记录该数据的有效性。数据的有效性是以所述数据是否被修改来确定的。如果所述数据是第一次写入，可以将所述数据记录为有效（称为有效数据）。如果所述数据被修改，则将所述修改前的数据记录为无效（称为无效数据）。具体的，可以用位图记录数据的有效性。例如，位图的每个“位”对应大小为 1KB 的数据的逻辑地址，当“位”为 1 时，代表

20 所述逻辑地址中存储的数据有效，当“位”为 0 时，代表所述逻辑地址中存储的数据无效。

系统垃圾回收是指控制器以段为单位执行垃圾回收。控制器根据位图获取一个段中有效数据的逻辑地址，将所述有效数据的逻辑地址发送给固态硬盘，并且所述控制器为所述有效数据分配新的逻辑地址，发送给所述固态硬盘

25 盘。所述固态硬盘从旧的逻辑地址中读取所述有效数据，写入所述新的逻辑地址中。然后，控制器再给固态硬盘发送去映射（英文：unmap）命令，所

述去映射命令包括所述待回收的段的逻辑地址区间，固态硬盘接收所述去映射命令之后，删除闪存翻译层中保存的所述逻辑地址区间的逻辑地址与实际地址之间的对应关系。固态硬盘还可以将所述实际地址对应的块标记为不包含有效数据的块。

- 5 系统垃圾回收有别于固态硬盘内部的垃圾回收。固态硬盘内部的垃圾回收是由固态硬盘自己完成的，不需要控制器配合。固态硬盘控制器先将各个块中的有效数据复制到空白的块，再擦除原来的块。另外，系统垃圾回收的触发条件通常是存储系统中空闲的段的数量不足（例如，低于某个阈值），而固态硬盘内部的垃圾回收的触发条件是该固态硬盘的空白的块的数量不足。
- 10 足。

段通常具有四种状态，分别是空闲（英文：free）、正在写入（英文：writing）、写满（英文：full）和损坏（英文：bad）。空闲状态的段是指无内容的段。正在写入状态的段是指已写入数据但还没有被写满的段。写满状态的段是指全部空间均已被写满的段。正在写入状态的段和写满状态的段经过

15 擦除之后可以成为空闲状态的段。损坏状态的段是指发生损坏不能使用的段。

下面结合图 3 介绍本实施例提供的写数据的方法，该方法可以应用于图 1 所示的应用场景中，由图 1 所示的控制器 11 或者控制器 22 执行。示例性的，该方法包括以下步骤：

- 20 S301，控制器 11 接收主机的数据，将数据写入控制器 11 的存储器 111。

具体的，可以由处理器 112 通过接口卡 110 接收主机的数据，将其写入存储器 111 中。通常情况下，所述数据是携带在写数据指令中发送给控制器 11 的。

- 25 S302，当存储器 111 中的数据的大小达到预先设定的阈值时，控制器 11 查找正在写入状态的段。由于存储器 111 的容量有限，当其中存储的数据达到所述预先设定的阈值时就需要将存储器 111 中的数据写入固态硬盘 44。

具体的，控制器 11 查找是否存在正在写入状态的段。例如，逻辑地址区间为 0KB-1023KB 的段，其中 0KB-100KB 的逻辑地址已写入数据，而 101KB 至 1023KB 的逻辑地址仍然没有被写入数据。那么控制器 11 就可以从 101KB 开始为所述数据分配逻辑地址。如果所述数据的大小为 2KB，那么控制器 11 为所述数据分配的起始逻辑地址为 101KB，长度为 2KB。

若当前不存在已经写入数据但未被写满的段，则执行步骤 S303。

S303，控制器 11 查找空闲的段，从所述空闲的段中为所述数据分配逻辑地址，将所述分配的逻辑地址发送给固态硬盘 44。例如，逻辑地址区间为 1024KB-2047KB 的段是空闲的段。控制器 11 可以从 1024KB 开始为所述数据分配逻辑地址。如果所述数据的大小为 2KB，那么控制器 11 为所述数据分配的逻辑地址为 1024KB-1026KB KB。

S304，控制器 11 将所述数据以及分配的逻辑地址发送给固态硬盘 44。

S305，固态硬盘 44 将所述数据连续写入一个或多个块。所述数据保存在块中的地址为实际地址。固态硬盘 44 在将所述数据写入所述块之后，将所述分配的逻辑地址与所述实际地址之间的对应关系保存在闪存翻译层中。

当一个段被控制器分配之后，这个段与这个控制器之间的归属关系也就确定了。例如控制器 11 选择空闲的逻辑地址区间为 1024KB-2047KB 的段，并向该段写入数据，那么该段就只能接收控制器 11 写入的数据，不能接收控制器 22 写入的数据。另外，当一个空闲的段被控制器 11 分配出来时，控制器 11 还可以为所述空闲的段分配标识。所述标识用以唯一识别所述分配的段。然而，固态硬盘控制器在将数据写入块时，不会区分所述数据是来自哪个控制器，只是按照接收数据的时间顺序连续写入所述块中。例如，控制器 11 将大小为 2KB，逻辑地址为 1024KB-1026KB 的数据发送给固态硬盘 44。固态硬盘控制器将该数据写入一个空白的块中，假设块的容量为 1M，那么这个块还有 1022KB 的存储空间可以存储数据。此时，如果控制器 22 也向固态硬盘 44 发送一个写数据请求。所述写数据请求包括大小为 3KB 的

数据，该数据的逻辑地址为 2048KB-2051KB。该数据对应的段是逻辑地址区间为 2048 KB -3071 KB 的段，并且该段是归属于控制器 22 的（控制器 22 分配的）。那么，固态硬盘控制器也将所述数据写入上述块中。此时，该块仍有 1019KB 的存储空间可以存储数据。如果还有其他控制器向固态硬盘 44 发送写数据请求，固态硬盘控制器会继续将其他控制器（也就是其他段）的数据写入所述块。或者，固态硬盘控制器也可以将控制器 11 发送的另一个段的数据（如果上一个段已被写满）写入所述块中。因此，一个块中存储的数据可能属于不同的段。

段的标识可以是数字、字母或者其他用于唯一标识所述段的符号，也可以是数字、字母或者其他符号之间的任意组合。如果控制器不分配段的标识，控制器可以用段的逻辑地址区间识别各个段。另外，所述标识也可以是序列号。在控制器 11 分配了标识之后，保存所述标识与所述段的逻辑地址区间之间的对应关系。后续将详细介绍为段分配标识的过程。

下面介绍控制器为段分配标识的过程。

控制器按照时间段为段分配标识（称为段标识）。在某些应用场景中，存储系统中的每个控制器（例如控制器 11 和控制器 22）都具有分配标识的功能，在另外一些应用场景中，存储系统中只有一个控制器具有分配标识的功能，在这种情况下不具有分配功能的控制器可以给具有该功能的控制器发送申请段标识的请求，以获得标识。

示例一，控制器 11 和控制器 22 都具有分配标识的功能。当控制器 11 分配一个空闲的段时，控制器 11 为该段分配标识。同样的，当控制器 22 分配另一个空闲的段时，控制器 22 也为所述另一个空闲的段分配标识。在相同时间段内分配标识的段都属于同一个段组，所述相同时间段的结束时间点为：任意一个控制器分配的标识的总数达到预设阈值的时刻。例如，所述预设阈值为 100，当控制器 11 分配的标识的总数达到 100 时，控制器 11 可以向控制器 22 发送通知消息，以指示控制器 22 当前时间段结束。那么在这段

时间内（这段时间的起点是控制器 11 或控制器 22 分配第一个标识，终点是控制器 11 分配第 100 个标识）控制器 11 为其分配标识的段，以及控制器 22 为其分配标识的段属于同一个组（称为段组）。控制器 11 或控制器 22 为该组分配组标识，并将所述组标识通知对方。控制器 11 保存所述组标识与该组中由控制器 11 管理的各个段的标识之间的对应关系，控制器 22 保存所述组标识与该组中由控制器 22 管理的各个段的标识之间的对应关系。

示例二，只有控制器 11 具有分配标识的功能。在这种情况下，当控制器 22 分配一个空闲的段时，控制器 22 会向控制器 11 发送请求消息，以请求控制器 11 为该段分配标识。此时，所述相同时间段的结束时间点为：具有标识分配功能的控制器分配的标识的总数达到预设阈值的时刻。例如，所述预设阈值为 100，在控制器 11 分配第 1 个标识至控制器 11 分配第 100 个标识的时间段内，控制器 11 为其分配标识的段属于同一个组。这个组中的段既有控制器 11 管理的段，也有控制器 22 管理的段，只是这些段的标识都是控制器 11 分配的。控制器 11 再为这个组分配组标识，并将所述组标识通知控制器 22。控制器 11 保存所述组标识与该组中由控制器 11 管理的各个段的标识之间的对应关系，控制器 22 保存所述组标识与该组中由控制器 22 管理的各个段的标识之间的对应关系。

在示例一和示例二中，都分配组标识，各个控制器记录组标识和段的标识之间的对应关系。如果不分配段的标识，并用段的逻辑地址区间识别各个段，控制器也分配组标识，并记录组标识和段的逻辑地址区间之间的对应关系。如果用序列号根据特定的规律标识段，组标识不是必需的。此时，可以以序列号的规律来判定哪些段属于同一个组。

例如，在示例三中，控制器 11 和控制器 22 都具有分配序列号的功能，并且控制器 11 和控制器 22 可以预先约定每个组的起始序列号。例如，第一组的起始序列号为 1，第二个组的起始序列号为 101，第三个组的起始序列号为 201，等等。控制器 11 分配的起始序列号与控制器 22 分配的起始序列

号相同。每个控制器分别从起始序列号开始，依次为段分配序列号，每个序列号等于上一个序列号的数值加 1。对于每个组，当任意一个控制器分配的序列号的总数达到预设阈值时，所有控制器都从下一个组的起始序列号开始分配。除了段的标识为序列号之外，示例三的其余内容与示例一类似，这里不再赘述。

示例四，只有控制器 11 具有分配标识的功能，并且所述标识是指序列号。以控制器 11 具有分配序列号的功能为例。控制器 11 预先设置每个组的起始序列号，从起始序列号开始，依次为自己的段以及控制器 22 的段分配序列号，每个序列号等于前一个序列号的数值加 1。当控制器 11 分配的序列号的总数达到预设阈值时，则从下一个组的起始序列号开始分配。具体的，当控制器 22 需要为自己的段分配序列号时，控制器 22 向控制器 11 发送请求消息，控制器 11 根据所述请求消息分配序列号，并发送给控制器 22。与示例三不同之处在于，在示例三中，由于控制器 11 和控制器 22 都可以为段分配序列号，因此这些序列号之间并不总是连续的，甚至可能会重复。而在示例四中，由于所有的序列号都是由控制器 11 分配的，所以这些序列号是连续的，并且不会重复。除了段的标识为序列号之外，示例四的其余内容与示例二类似，这里不再赘述。

由上面的描述可知，固态硬盘的一个块中存储的数据属于不同的段。假设一个块存储的数据分别属于序列号为 1 的段，序列号为 2 的段以及序列号为 3 的段。如果控制器对序列号为 1 的段以及序列号为 2 的段进行垃圾回收，固态硬盘将这两个段对应的有效的数据复制到其他段中。然而，对于上述块来说，仍然保留有序列号为 3 的段的数据。因此，当固态硬盘进行自己的垃圾回收时，仍然需要从该块中读取有效数据，并复制到其他块。从而导致固态硬盘内部出现写放大。

为了减少固态硬盘内部的写放大的次数，本发明实施例在执行系统垃圾回收操作时，以上面描述的段组为单位对段进行垃圾回收。由于该段组中的

各个段是在相同时间段内分配的，那么这些段的数据也是在相同的时间段内写入固态硬盘的。进一步地，由于固态硬盘在写入数据时是按照接收数据的时间顺序连续写入一个或多个块，因此，相同的时间段内分配的段被执行系统垃圾回收之后，这些段所对应的块中的有效数据已经被全部复制到其他块中，这些段所对应的块中的数据都是无效数据。当固态硬盘进行内部的垃圾回收时，被执行系统垃圾回收的段所对应的块无需再次进行有效数据的复制，从而减少了固态硬盘内部的写放大的次数。

下面介绍本实施例提供的系统垃圾回收的方法。

本发明实施例提供了一种系统垃圾回收的方法，应用在图 1 所示的存储系统中，所述存储系统包含多个段，其中部分段属于控制器 11，部分段属于控制器 22，所述存储系统包含的多个段可以划分为若干个段组。其中，每个段组中的段是在相同的时间段内分配的。每个段组具有唯一的组标识。本实施例提供的系统垃圾回收是以段组为单位来执行的。示例性的，该方法可以包括以下步骤：

401，控制器 11 统计各个段组包含的无效数据的数据量。

当空闲的段的数量低于预设阈值时，控制器 11 可以触发系统垃圾回收操作。在执行系统垃圾回收之前，控制器 11 可以统计每个段组包含的无效数据的数据量。由于一个段组所包含的各个段中，既有控制器 11 管理的段，又有控制器 22 管理的段（这里是以两个控制器为例，但本实施例还可以包括其他控制器）。一种实现是，控制器 11 只统计每个组中自己管理的那些段的无效数据的数据量，并且将包含无效数据最多的段组作为当前待回收的组。另一种实现是，控制器 11 不仅需要统计每个组中自己管理的那些段的无效数据的数据量，还需要获取每个组中其他控制器管理的那些段的无效数据的数据量。具体的，控制器 11 可以向其他控制器（例如控制器 22）发送查询指令，所述查询指令包括组标识，控制器 22 根据所述组标识查询该段组所包括的段的无效数据的数据量，并反馈给控制器 11。

另外，由前面的描述可知，当数据写入段之后，可以用位图来记录所述数据是否有效。因此，根据所述位图也可以统计一个段所包括的无效数据的数据量。

402，控制器 11 比较各个段组包含的无效数据的数据量，确定包含无效数据最多的段组。

控制器 11 根据自己统计的数据量，或者根据自己统计的数据量以及控制器 22 统计的数据量，可以得出哪个段组所包括的无效数据的数据量最多，将其作为当前待回收的段组。

403，控制器 11 对所述确定出的段组所包含的段执行系统垃圾回收操作。

具体的，在步骤 403 中控制器 11 指示固态硬盘 44 对所述确定出的段组所包含的段执行系统垃圾回收操作。这里的“指示”是指控制器 11 向固态硬盘 44 发送读数据请求、写数据请求以及去映射命令，使得固态硬盘 44 执行这些请求及命令。可以理解的是，固态硬盘 44 并不会主动地执行系统垃圾回收操作，只是被动地执行控制器 11 发送的请求及命令。控制器 11 指示固态硬盘 44 对所述确定出的段组所包含的段执行系统垃圾回收操作具体包括下述步骤：（1）控制器 11 向固态硬盘 44 发送读数据请求，所述读数据请求包括源逻辑地址。所述源逻辑地址是指一个段中有效数据的逻辑地址。固态硬盘 44 接收所述读数据请求之后，从所述源逻辑地址对应的实际地址中读取有效数据，写入固态硬盘控制器的存储器中。（2）控制器 11 向固态硬盘 44 发送写数据请求，所述写数据请求包括目标逻辑地址。所述目标逻辑地址是指控制器 11 为所述有效数据重新分配的逻辑地址。并且，目标逻辑地址与源逻辑地址对应不同的段，其中源逻辑地址对应的段是待回收的段，而目标逻辑地址对应的段是一个空闲的段或者处于正在写入状态的段。固态硬盘 44 接收所述写数据请求之后，从固态硬盘控制器的存储器中读取所述有效数据，并写入所述目标逻辑地址对应的实际地址中。（3）控制器 11 向固态硬盘 44 发送去映射命令，所述去映射命令包括所述待回收的段的逻辑

地址区间，固态硬盘 44 接收所述去映射命令之后，在闪存翻译层中删除所述逻辑地址区间中的每个逻辑地址与实际地址之间的对应关系。

404，控制器 11 将所述段组的组标识发送给控制器 22，控制器 22 对所述段组所包含的段执行系统垃圾回收操作。

- 5 控制器 22 根据所述组标识，以及所述组标识与段的标识之间的对应关系，确定待回收的段，对所述待回收的段执行系统垃圾回收操作，具体的过程与 403 类似，这里不再赘述。

- 步骤 403 中控制器 11 执行系统垃圾回收的段与步骤 404 中控制器 22 执行系统垃圾回收的段是在相同的时间段内分配的。所述相同的时间段在所述
10 第一控制器或所述第二控制器在所述时间段内分配的段的数量达到第一预设阈值时结束。当每个段都分配有标识并且每个控制器都具有分配标识的功能时，这里的相同的时间段与上面描述的示例一中的时间段一致。

- 另外，所述相同的时间段也可以在所述第一控制器以及第二控制器在所述时间段内分配的段的数量的总和达到第二预设阈值时结束。第二预设阈值
15 可以与第一预设阈值相同，也可以不同。在这种情况下，第一控制器或者第二控制器需要给对方发送消息以查询对方分配的段的数量，从而获得这段时间内分配的段的数量的总和。当每个段都分配有标识并且只有一个控制器都具有分配标识的功能时，这里描述的相同的时间段也可以在具有分配标识的功能的控制器分配的标识的数量达到第二预设阈值时结束。

- 20 除了由控制器 11 统计各个段组包含的无效数据的数据量之外，本实施例还有一种实现是，在存储系统中指定一个控制器，专门用于统计各个段组包含的无效数据的数据量，例如，所述指定的控制器分别向控制器 11 和控制器 22 发送查询指令，以获取每个段组中控制器 11 管理的段包含的无效数据的数据量，以及每个段组中控制器 22 管理的段包含的无效数据的数据量。
25 根据查询结果，确定包含无效数据最多的段组，并将所述段组的组标识分别发送给控制器 11 和控制器 22。控制器 11 和控制器 22 分别根据所述组标识

与段的标识之间的对应关系，确定待回收的段，对所述待回收的段执行系统垃圾回收操作。

另外，本实施例也可以不对各个段组所包含的无效数据的数据量进行比较，而是监控每个段组所包含的无效数据的数据量，当某一个段组所包含的
5 无效数据的数据量达到预设阈值时，对所述无效数据的数据量达到预设阈值的段组执行垃圾回收操作。

或者，控制器还可以依次对每个段组执行系统垃圾回收操作，不考虑段组中无效数据的大小。

或者，控制器还可以根据每个段组上一次执行系统垃圾回收操作的时间，选择最长时间未进行系统垃圾回收的段组作为当前系统垃圾回收操作的对象。
10

在上述步骤 404 中，控制器 22 是在控制器 11 执行系统垃圾回收时，对所述确定出的段组中由控制器 22 分配的段执行系统垃圾回收操作的。但是，控制器 11 和控制器 22 在执行系统垃圾回收操作时，并不是绝对同时进行的。
15 只要在启动下一个段组的系统垃圾回收之前，控制器 11 和控制器 22 都完成了所述确定出的段组的系统垃圾回收即可。在实际应用中，控制器 11 和控制器 22 是以段组为单位来执行系统垃圾回收操作的。当一个段组中所包含的所有段都完成系统垃圾回收操作之后，再启动下一个段组的系统垃圾回收操作。这里的段组中所包含的所有段都完成系统垃圾回收操作是指，控制器
20 11 完成其管理的该段组中的段的系统垃圾回收操作，并且控制器 22 也完成了其管理的段组中的段的系统垃圾回收操作。示例性的，当控制器 11 和控制器 22 完成对包含无效数据最多的段组的系统垃圾回收时，再查找下一个包含无效数据最多的段组并启动系统垃圾回收。

本领域普通技术人员将会理解，本发明的各个方面、或各个方面的可能
25 实现方式可以被具体实施为系统、方法或者计算机程序产品。因此，本发明的各方面、或各个方面的可能实现方式可以采用完全硬件实施例、完全软件

实施例 (包括固件、驻留软件等等), 或者组合软件和硬件方面的实施例的形式, 在这里都统称为“电路”、“模块”或者“系统”。此外, 本发明的各方面、或各个方面的可能实现方式可以采用计算机程序产品的形式, 计算机程序产品是指存储在计算机可读介质中的计算机可读程序代码。

- 5 计算机可读介质包含但不限于电子、磁性、光学、电磁、红外或半导体系统、设备或者装置, 或者前述的任意适当组合, 如随机访问存储器(RAM)、只读存储器(ROM)、可擦除可编程只读存储器(EPROM)、光盘。

 计算机中的处理器读取存储在计算机可读介质中的计算机可读程序代码, 使得处理器能够执行在流程图中每个步骤、或各步骤的组合中规定的功
10 能动作。

 计算机可读程序代码可以完全在用户的计算机上执行、部分在用户的计算机上执行、作为单独的软件包、部分在用户的计算机上并且部分在远程计算机上, 或者完全在远程计算机或者服务器上执行。也应该注意, 在某些替代实施方案中, 在流程图中各步骤、或框图中各块所注明的功能可能不按图中注明的顺序发生。例如, 依赖于所涉及的功能, 接连示出的两个步骤、或
15 两个块实际上可能被大致同时执行, 或者这些块有时候可能被以相反顺序执行。

 本领域普通技术人员可以意识到, 结合本文中所公开的实施例描述的各示例的单元及算法步骤, 能够以电子硬件、或者计算机软件和电子硬件的结合来实现。这些功能究竟以硬件还是软件方式来执行, 取决于技术方案的特
20 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发明的范围。

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

权利要求

1、一种存储系统，其特征在于，包括第一控制器、第二控制器和固态硬盘，所述固态硬盘和所述第一控制器连接，所述固态硬盘和所述第二控制器连接，所述第一控制器或第二控制器以段为单位管理所述固态硬盘的存储空间，其中，

所述第一控制器，用于对所述第一控制器管理的段中的多个段进行系统垃圾回收，所述第一控制器管理的段中的多个段是在一个时间段内由所述第一控制器分配的；并且，

所述第二控制器，用于当所述第一控制器进行系统垃圾回收时，对所述第二控制器管理的段中的多个段进行系统垃圾回收，所述第二控制器管理的段中的多个段是在所述时间段内由所述第二控制器分配的。

2、根据权利要求 1 所述的存储系统，其特征在于，所述第一控制器管理的段中的多个段是所述时间段内由所述第一控制器分配的所有的段，所述第二控制器管理的段中的多个段是所述时间段内由所述第二控制器分配的所有的段。

3、根据权利要求 1 或 2 所述的存储系统，其特征在于，所述时间段在所述第一控制器或所述第二控制器在所述时间段内分配的段的数量达到第一预设阈值时结束。

4、根据权利要求 1-3 任一所述的存储系统，其特征在于，所述第一控制器管理的段中的多个段与所述第二控制器管理的段中的多个段属于同一个段组，所述段组包含的无效数据的数据量大于其他任意一个段组包含的无效数据的数据量。

5、根据权利要求4所述的存储系统，其特征在于，

所述第一控制器，还用于将所述段组的组标识发送给所述第二控制器；

所述第二控制器，还用于根据预先保存的所述组标识与所述第二控制器管理的段中的多个段中每个段的标识之间的对应关系，确定所述第二控制器管理的段中的所述多个段。

6、根据权利要求1-5任一所述的存储系统，其特征在于，

所述第一控制器，用于将所述第一控制器管理的段中的多个段的每个段包含的有效数据的源逻辑地址发送给所述固态硬盘；为所述有效数据分配目标逻辑地址并将所述目标逻辑地址发送给所述固态硬盘，以指示所述固态硬盘将所述有效数据从所述源逻辑地址复制到所述目标逻辑地址；向所述固态硬盘发送去映射命令，所述去映射命令包括所述第一控制器管理的段中的多个段的每个段的逻辑地址区间，以指示所述固态硬盘删除所述固态硬盘保存的所述每个段的逻辑地址与实际地址之间的对应关系。

7、一种系统垃圾回收方法，其特征在于，所述方法应用于存储系统中，所述存储系统包括第一控制器、第二控制器和固态硬盘，所述固态硬盘和所述第一控制器连接，所述固态硬盘和所述第二控制器连接，所述第一控制器或第二控制器以段为单位管理所述固态硬盘的存储空间，所述方法包括：

所述第一控制器对所述第一控制器管理的段中的多个段进行系统垃圾回收，所述第一控制器管理的段中的多个段是在一个时间段内由所述第一控制器分配的；

当所述第一控制器进行系统垃圾回收时，所述第二控制器对所述第二控制器管理的段中的多个段进行系统垃圾回收，所述第二控制器管理的段中的多个段是在所述时间段内由所述第二控制器分配的。

8、根据权利要求 7 所述的方法，其特征在于，所述第一控制器管理的段中的多个段是所述时间段内由所述第一控制器分配的所有的段，所述第二控制器管理的段中的多个段是所述时间段内由所述第二控制器分配的所有
5 的段。

9、根据权利要求 7 或 8 所述的方法，其特征在于，所述时间段在所述第一控制器或所述第二控制器在所述时间段内分配的段的数量达到第一预设阈值时结束。

10

10、根据权利要求 7-9 任一所述的方法，其特征在于，所述第一控制器管理的段中的多个段与所述第二控制器管理的段中的多个段属于同一个段组，所述段组包含的无效数据的数据量大于其他任意一个段组包含的无效数据的数据量。

15

11、根据权利要求 10 所述的方法，其特征在于，所述方法还包括：

所述第一控制器将所述段组的组标识发送给所述第二控制器；

所述第二控制器根据预先保存的所述组标识与所述第二控制器管理的段中的多个段中每个段的标识之间的对应关系，确定所述第二控制器管理的
20 段中的所述多个段。

12、根据权利要求 7-11 任一所述的方法，其特征在于，

所述第一控制器对所述第一控制器管理的段中的多个段进行系统垃圾回收包括：

25 所述第一控制器将所述第一控制器管理的段中的多个段的每个段包含的有效数据的源逻辑地址发送给所述固态硬盘；

所述第一控制器为所述有效数据分配目标逻辑地址并将所述目标逻辑地址发送给所述固态硬盘，以指示所述固态硬盘将所述有效数据从所述源逻辑地址复制到所述目标逻辑地址；

- 5 所述第一控制器向所述固态硬盘发送去映射命令，所述去映射命令包括所述第一控制器管理的段中的多个段的每个段的逻辑地址区间，以指示所述固态硬盘删除所述固态硬盘保存的所述每个段的逻辑地址与实际地址之间的对应关系。