

一种数据校验的方法及存储系统

技术领域

本发明实施例涉及存储技术领域，特别是一种数据校验的方法及存储系统。

5 背景技术

在包含多个存储节点的分布式系统中，主机将生成的多个写数据请求发送给多个存储节点，每个存储节点各自存储一些写数据请求。写数据请求中包括数据以及所述数据待写入存储节点的逻辑地址（以下简称为逻辑地址）。为了防止某个存储节点发生故障导致其存储的写数据请求丢失，所述主机会
10 计算所述生成的多个写数据请求的校验数据，并将校验数据发送给一个或多个存储节点存储。为了计算校验数据，主机将达到设定大小的多个写数据请求按照预定尺寸进行划分，得到多个数据单元，再计算所述多个数据单元的校验单元。这些数据单元和校验单元组成一个分条。最后，主机将每个数据单元或者校验单元发送给一个存储节点存储。由于这些写数据请求被随机地
15 划分在多个数据单元中，从而发送给不同的存储节点存储。因此，在读取数据的时候，往往需要从不同的存储节点收集多个写数据请求携带的数据。跨存储节点的读取操作会影响读取数据的效率。

20 发明内容

本申请第一方面提供了一种存储系统。存储系统包括主机、校验节点和多个数据节点。每个数据节点具有唯一的标识。所述主机用于根据写数据请求中包括的数据节点的标识，将多个写数据请求划分为多个写数据请求集合。每个写数据请求集合包括具有相同的数据节点的标识的一个或多个写数
25 据请求。每个写数据请求包括数据、所述数据待写入的逻辑地址以及所述数据待写入的数据节点的标识。所述数据待写入的数据节点是所述主机根据所

述数据或所述逻辑地址选择的，或者，用户在主机 11 中触发多个写数据请求时，向主机 11 发送指令要求将这些写数据请求写入同一个数据节点。主机 11 可以根据用户的需求，为这些写数据请求选择一个数据节点，将这个数据节点携带在每个写数据请求。当设定数量个写数据请求集合中的每个写数据请求集合中的所有写数据请求的大小达到预设数据量时，所述主机计算所述设定数量个写数据请求集合的校验数据。所述设定数量个写数据请求集合是所述多个写数据请求集合的子集。所述主机将所述设定数量个写数据请求集合中的每个写数据请求集合发送给所述写数据请求集合包括的数据节点的标识所指示的数据节点，将所述校验数据发送给所述校验节点。

10 在本申请提供的存储系统中，主机按照写数据请求包含的数据节点的标识，将多个写数据请求划分为多个写数据请求集合。当设定数量个写数据请求集合中所有写数据请求的大小达到预设数据量时，主机计算所述设定数量个写数据请求集合的校验数据，并将校验数据发送给校验节点存储，由此保证了数据的可靠性。并且，由于每个写数据请求集合包含待写入相同的数据节点 15 的多个写数据请求，而每个写数据请求待写入的数据节点是所述主机根据所述写数据请求中的数据或者所述数据待写入的逻辑地址选择的。因此，每个写数据请求集合中保存的是相同类别的写数据请求。当主机计算校验数据之后，将每个写数据请求集合发送给所述写数据请求集合包含的标识所指示的数据节点，相同类别的写数据请求就被存储在了同一个数据节点中。由于 20 于相同类别的写数据请求中的数据被同时读取的可能性较大，那么在这些数据被读取时可以在一个数据节点内读取，不用跨节点执行读取操作，提高了数据读取的效率。

结合第一方面，在第一方面的第一种实现中，所述主机还用于为所述设定数量个写数据请求集合中的每个写数据请求分配标识，并且将每个写数据请求集合的标识发送给该写数据请求集合包含的数据节点的标识所指示的数据节点。所述数据节点用于接收写数据请求集合以及所述写数据请求集

合的标识。所述数据节点还用于创建并保存元数据。所述元数据包括所述写数据请求集合的标识与每个写数据请求中的数据待写入的逻辑地址之间的对应关系，以及每个写数据请求中的数据待写入的逻辑地址与内部偏移量之间的对应关系。在本申请中，由于相同类别的写数据请求作为一个写数据请求集合保存在一个数据节点，那么与这个写数据请求集合相关的元数据也保存在数据节点本地。后续如果对写数据请求集合进行垃圾回收等操作，可以直接在本地对元数据进行修改。

结合第一方面的第一种实现，在第一方面的第二种实现中，所述存储系统还包括元数据校验计算节点以及元数据校验节点。所述数据节点还用于在确定累积的元数据达到所述预设数据量时，将元数据集合以及所述数据节点的标识发送给所述元数据校验计算节点。所述元数据集合包括所述累积的达到所述预设数据量的元数据。所述元数据校验计算节点用于接收多个数据节点中每个数据节点发送的元数据集合以及所述数据节点的标识，保存每个元数据集合与数据节点的标识之间的对应关系。并且，根据所述对应关系从接收的多个元数据集合中选择设定数量个元数据集合。所述选择出的设定数量个元数据集合对应不同的数据节点的标识。所述元数据校验计算节点还用于计算所述选择出的设定数量个元数据集合的校验数据。然后，所述元数据校验计算节点将所述设定数量个元数据集合的校验数据发送给所述元数据校验节点。所述元数据校验节点不同于存储所述设定数量个元数据集合中的每个元数据集合所在的数据节点。由此，本申请提供的存储系统保证了各个数据节点中存储的元数据集合的可靠性。

结合第一方面的第一种实现，在第一方面的第三种实现中，所述存储系统还包括垃圾回收节点。垃圾回收节点用于执行系统垃圾回收操作。具体的，垃圾回收节点根据所述主机发送的分条标识与所述分条包括的每个写数据请求集合的标识之间的对应关系，以及数据节点发送的写数据请求集合的位图从多个分条中选择出包含无效数据最多的分条。本申请中的分条包括设定

数量个写数据请求集合以及根据这些写数据请求集合并计算出的校验数据。本申请中的位图用于指示写数据请求集合包含的无效数据的数据量。所述垃圾回收节点将所述包含无效数据最多的分条作为待回收的分条，向所述待回收的分条所包含的每个写数据请求集合所在的数据节点分别发送垃圾回收通

- 5 知消息。每个垃圾回收通知消息用于通知数据节点进行系统垃圾回收，每个垃圾回收通知消息中包括写数据请求集合的标识。由于垃圾回收单元在执行系统垃圾回收操作时会选择包含无效数据的分条，因此提高了系统垃圾回收的效率。

- 结合第一方面的第三种实现，在第一方面的第四种实现中，数据节点在
10 接收垃圾回收通知消息之后，根据写数据请求集合的标识以及保存的所述写数据请求集合的位图，对所述写数据请求集合执行系统垃圾回收操作。具体的，数据节点根据所述写数据请求集合的标识确定待回收的写数据请求集合。数据节点再根据所述写数据请求集合的位图确定所述写数据请求集合中包含的有效数据待写入的逻辑地址，删除所述有效数据待写入的逻辑地址与
15 所述写数据请求集合的标识之间的对应关系，并且保存所述有效数据待写入的逻辑地址与重新拼凑的写数据请求集合的标识之间的对应关系。然后，数据节点将所述写数据请求集合的硬盘逻辑地址发送给固态硬盘，由固态硬盘将与所述硬盘逻辑地址对应的块标记为无效。当后续固态硬盘进行内部的垃圾回收时可以直接擦除这些块，无需再次进行有效数据的复制，减少了固态硬盘内部写放大的次数。另外，本申请提供的系统垃圾回收是由每个数据节点将有效数据拼凑在本地的其他写数据请求集合中，因此所述数据节点独立完成系统垃圾回收，无需和其他数据节点进行数据交互，节省了数据节点之间的带宽。
20

- 本申请第二方面提供了一种数据校验方法，应用在第一方面或者第一方面
25 面的任意一种实现的存储系统中。

本申请第三方面提供了一种主机，所述主机的功能与第一方面的第一种

实现所提供的存储系统一致。

本申请第四方面提供了一种存储系统。所述存储系统包括校验节点、校验计算节点和多个数据节点。每个数据节点，用于将写数据请求集合发送给所述校验计算节点。所述写数据请求集合包括一个或多个写数据请求，每个写数据请求包括数据以及所述数据待写入的数据节点的标识。所述写数据请求集合的大小等于预设数据量。所述校验计算节点，用于接收多个写数据请求集合，从所述多个写数据请求集合中选择设定数量个写数据请求集合。所述设定数量个写数据请求集合是所述多个写数据请求集合的子集，并且所述设定数量个写数据请求集合包括不同的数据节点的标识。然后，所述校验计算节点计算所述设定数量个写数据请求集合的校验数据，将所述校验数据发送给所述校验节点。所述校验节点不同于所述设定数量个写数据请求集合中的每个写数据请求集合所在的数据节点。

本申请提供的数据校验方法，每个数据节点在累积的所有写数据请求的大小达到预设数据量时，将这些写数据请求作为一个写数据请求集合发送给校验计算节点。校验计算节点从接收的多个写数据请求集合中选择设定数量个写数据请求集合计算所述设定数量个写数据请求集合的校验数据，并将校验数据发送给校验节点存储，由此保证了数据的可靠性。

结合第四方面，在第四方面的第一种实现中，每个写数据请求还包括所述数据待写入的逻辑地址，所述数据待写入的数据节点是所述存储系统中的主机根据所述数据或者所述数据待写入的逻辑地址选择的。或者，所述数据待写入的数据节点也可能不是所述主机选择的，而是用户在所述主机中触发写数据请求时直接指定的。每个数据节点，还用于接收所述主机发送的多个写数据请求。由于每个写数据请求集合包含具有相同的数据节点的标识的多个写数据请求，而每个写数据请求待写入的数据节点是所述主机根据所述写数据请求中的数据或者所述数据待写入的逻辑地址选择的。因此，每个写数据请求集合中保存的是相同类别的写数据请求。由于相同类别的写数据请求

中的数据被同时读取的可能性较大，那么在这些数据被读取时可以在一个数据节点内读取，不用跨节点执行读取操作，提高了数据读取的效率。

结合第四方面的第一种实现，在第四方面的第二种实现中，所述数据节点，还用于为保存的写数据请求集合分配标识，将所述写数据请求集合的标识发送给所述校验计算节点。然后，所述数据节点创建并保存元数据。所述元数据包括所述写数据请求集合的标识与每个写数据请求中的数据待写入的逻辑地址之间的对应关系，以及每个写数据请求中的数据待写入的逻辑地址与内部偏移量之间的对应关系。在本申请中，由于相同类别的写数据请求作为一个写数据请求集合保存在一个数据节点，那么与这个写数据请求集合相关的元数据也保存在数据节点本地。后续如果对写数据请求集合进行垃圾回收等操作，可以直接在本地对元数据进行修改。

结合第四方面的第二种实现，在第四方面的第三种实现中，所述存储系统还包括元数据校验计算节点以及元数据校验节点。所述数据节点，还用于在确定累积的元数据达到所述预设数据量时，将元数据集合以及数据节点的标识发送给所述元数据校验计算节点。所述元数据集合包括所述累积的达到所述预设数据量的元数据。所述元数据校验计算节点，用于接收每个数据节点发送的元数据集合以及所述数据节点的标识，保存每个元数据集合与数据节点的标识之间的对应关系。并且，所述元数据校验计算节点根据所述对应关系从接收的多个元数据集合中选择所述设定数量个元数据集合。所述设定数量个元数据集合对应不同的数据节点的标识。所述元数据校验计算节点计算所述设定数量个元数据集合的校验数据，将所述设定数量个元数据集合的校验数据发送给所述元数据校验节点。所述元数据校验节点不同于存储所述设定数量个元数据集合中的每个元数据集合的数据节点。由此，本申请提供的存储系统保证了各个数据节点中存储的元数据集合的可靠性。

结合第四方面的第二种实现，在第四方面的第四种实现中，所述存储系统还包括垃圾回收节点。所述校验计算节点，还用于为分条分配标识。所述

分条包括所述设定数量个写数据请求集合以及所述设定数量个写数据请求集合的校验数据。然后，所述校验计算节点将所述分条标识与所述分条包括的每个写数据请求集合的标识之间的对应关系，发送给所述垃圾回收节点。每个数据节点，还用于将保存的写数据请求集合的位图发送给所述垃圾回收节点，所述位图用于指示所述写数据请求集合的无效数据的数据量。所述垃圾回收节点，用于根据所述分条标识与所述分条包括的每个写数据请求集合的标识之间的对应关系，以及每个写数据请求集合的位图从多个分条中选择包含无效数据最多的分条，向所述分条包含的每个写数据请求集合所在的数据节点分别发送垃圾回收通知消息。每个垃圾回收通知消息包括写数据请求集合的标识。由于垃圾回收单元在执行系统垃圾回收操作时会选择包含无效数据的数据的分条，因此提高了系统垃圾回收的效率。

结合第四方面的第四种实现，在第四方面的第五种实现中，数据节点在接收垃圾回收通知消息之后，根据写数据请求集合的标识以及保存的所述写数据请求集合的位图，对所述写数据请求集合执行系统垃圾回收操作。具体的，数据节点根据所述写数据请求集合的标识确定待回收的写数据请求集合。数据节点再根据所述写数据请求集合的位图确定所述写数据请求集合中包含的有效数据待写入的逻辑地址，删除所述有效数据待写入的逻辑地址与所述写数据请求集合的标识之间的对应关系，并且保存所述有效数据待写入的逻辑地址与重新拼凑的写数据请求集合的标识之间的对应关系。然后，数据节点将所述写数据请求集合的硬盘逻辑地址发送给固态硬盘，由固态硬盘将与所述硬盘逻辑地址对应的块标记为无效。当后续固态硬盘进行内部的垃圾回收时可以直接擦除这些块，无需再次进行有效数据的复制，减少了固态硬盘内部写放大的次数。另外，本申请提供的系统垃圾回收是由每个数据节点将有效数据拼凑在本地的其他写数据请求集合中，因此所述数据节点独立完成系统垃圾回收，无需和其他数据节点进行数据交互，节省了数据节点之间的带宽。

本申请第五方面提供了一种数据校验方法，应用在第四方面或者第四方面的任意一种实现的存储系统中。

本申请第六方面提供了一种存储系统。存储系统包括主机、校验计算节点和多个数据节点。每个数据节点具有唯一的标识。所述主机将生成的多个写数据请求发送给校验计算节点。每个写数据请求集合包括具有相同的数据节点的标识的一个或多个写数据请求。每个写数据请求包括数据、所述数据待写入的逻辑地址以及所述数据待写入的数据节点的标识。所述数据待写入的数据节点是所述主机根据所述数据或所述逻辑地址选择的。或者，所述数据待写入的数据节点也可能不是所述主机选择的，而是用户在所述主机中触发写数据请求时直接指定的。所述校验计算节点用于根据写数据请求中包括的数据节点的标识，将多个写数据请求划分为多个写数据请求集合。当设定数量个写数据请求集合中的每个写数据请求集合中的所有写数据请求的大小达到预设数据量时，所述校验计算节点计算所述设定数量个写数据请求集合的校验数据。所述设定数量个写数据请求集合是所述多个写数据请求集合的子集。所述校验计算节点将所述设定数量个写数据请求集合中的每个写数据请求集合发送给所述写数据请求集合包括的数据节点的标识所指示的数据节点。所述校验计算节点保存所述校验数据。

在本申请提供的存储系统中，校验计算节点按照写数据请求包含的数据节点的标识，将多个写数据请求划分为多个写数据请求集合。当设定数量个写数据请求集合中所有写数据请求的大小达到预设数据量时，校验计算节点计算所述设定数量个写数据请求集合的校验数据，并将校验数据发送给校验节点存储，由此保证了数据的可靠性。并且，由于每个写数据请求集合包含待写入相同的数据节点的多个写数据请求，而每个写数据请求待写入的数据节点是所述主机根据所述写数据请求中的数据或者所述数据待写入的逻辑地址选择的。因此，每个写数据请求集合中保存的是相同类别的写数据请求。当主机计算校验数据之后，将每个写数据请求集合发送给所述写数据请求集

合包含的标识所指示的数据节点，相同类别的写数据请求就被存储在了同一个数据节点中。由于相同类别的写数据请求中的数据被同时读取的可能性较大，那么在这些数据被读取时可以在一个数据节点内读取，不用跨节点执行读取操作，提高了数据读取的效率。结合第六方面，在第六方面的第一种实现中，所述校验计算节点还用于为所述设定数量个写数据请求集合中的每个写数据请求分配标识，并且将每个写数据请求集合的标识发送给该写数据请求集合包含的数据节点的标识所指示的数据节点。所述数据节点用于接收写数据请求集合以及所述写数据请求集合的标识。所述数据节点还用于创建并保存元数据。所述元数据包括所述写数据请求集合的标识与每个写数据请求中的数据待写入的逻辑地址之间的对应关系，以及每个写数据请求中的数据待写入的逻辑地址与内部偏移量之间的对应关系。在本申请中，由于相同类别的写数据请求作为一个写数据请求集合保存在一个数据节点，那么与这个写数据请求集合相关的元数据也保存在数据节点本地。后续如果对写数据请求集合进行垃圾回收等操作，可以直接在本地对元数据进行修改。

结合第六方面的第一种实现，在第六方面的第二种实现中，所述存储系统还包括元数据校验计算节点以及元数据校验节点。所述数据节点还用于在确定累积的元数据达到所述预设数据量时，将元数据集合以及所述数据节点的标识发送给所述元数据校验计算节点。所述元数据集合包括所述累积的达到所述预设数据量的元数据。所述元数据校验计算节点用于接收多个数据节点中每个数据节点发送的元数据集合以及所述数据节点的标识，保存每个元数据集合与数据节点的标识之间的对应关系。并且，根据所述对应关系从接收的多个元数据集合中选择设定数量个元数据集合。所述选择出的设定数据量个元数据集合对应不同的数据节点的标识。所述元数据校验计算节点还用于计算所述选择出的设定数量个元数据集合的校验数据。然后，所述元数据校验计算节点将所述设定数量个元数据集合的校验数据发送给所述元数据校验节点。所述元数据校验节点不同于存储所述设定数量个元数据集合中的

每个元数据集合所在的数据节点。由此，本申请提供的存储系统保证了各个数据节点中存储的元数据集合的可靠性。

结合第六方面的第一种实现，在第六方面的第三种实现中，所述存储系统还包括垃圾回收节点。垃圾回收节点用于执行系统垃圾回收操作。具体的，

5 垃圾回收节点根据所述校验计算节点发送的分条标识与所述分条包括的每个写数据请求集合的标识之间的对应关系，以及数据节点发送的写数据请求集合的位图从多个分条中选择出包含无效数据最多的分条。本申请中的分条包括设定数量个写数据请求集合以及根据这些写数据请求集合计算出的校验数据。本申请中的位图用于指示写数据请求集合包含的无效数据的数据

10 量。所述垃圾回收节点将所述包含无效数据最多的分条作为待回收的分条，向所述待回收的分条所包含的每个写数据请求集合所在的数据节点分别发送垃圾回收通知消息。每个垃圾回收通知消息用于通知数据节点进行系统垃圾回收，每个垃圾回收通知消息中包括写数据请求集合的标识。由于垃圾回收单元在执行系统垃圾回收操作时会选择包含无效数据的分条，因此提高了

15 系统垃圾回收的效率。

结合第六方面的第三种实现，在第六方面的第四种实现中，数据节点在接收垃圾回收通知消息之后，根据写数据请求集合的标识以及保存的所述写数据请求集合的位图，对所述写数据请求集合执行系统垃圾回收操作。具体的，数据节点根据所述写数据请求集合的标识确定待回收的写数据请求集

20 合。数据节点再根据所述写数据请求集合的位图确定所述写数据请求集合中包含的有效数据待写入的逻辑地址，删除所述有效数据待写入的逻辑地址与所述写数据请求集合的标识之间的对应关系，并且保存所述有效数据待写入的逻辑地址与重新拼凑的写数据请求集合的标识之间的对应关系。然后，数据节点将所述写数据请求集合的硬盘逻辑地址发送给固态硬盘，由固态硬盘

25 将与所述硬盘逻辑地址对应的块标记为无效。当后续固态硬盘进行内部的垃圾回收时可以直接擦除这些块，无需再次进行有效数据的复制，减少了固态

硬盘内部写放大的次数。另外，本申请提供的系统垃圾回收是由每个数据节点将有效数据拼凑在本地的其他写数据请求集合中，因此所述数据节点独立完成系统垃圾回收，无需和其他数据节点进行数据交互，节省了数据节点之间的带宽。

- 5 本申请第七方面提供了一种数据校验方法，应用在第七方面或者第七方面的任意一种实现的存储系统中。

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。

- 10 图 1 是本发明实施例提供的存储系统的组成图；
 图 2 是本发明实施例提供的主机的结构图；
 图 3 是本发明实施例提供的闪存阵列的结构图；
 图 4 是本发明实施例提供的一种数据校验方法的流程示意图；
 图 5 是本发明实施例提供的另一种数据校验方法的流程示意图；
15 图 6 是本发明实施例提供的再一种数据校验方法的流程示意图；
 图 7 是本发明实施例提供的系统垃圾回收方法的流程示意图。

具体实施方式

- 本发明实施例提出了一种数据校验的方法及存储系统，能够在保证数据的可靠性的基础上，将相同类别的数据存储在同一个存储节点中，在读取这
20 些数据时只需在一个存储节点内读取，从而提高读取数据的效率。

 图 1 描绘了本发明实施例提供的存储系统 10 的组成图，图 1 所示的存储系统 10 包括主机 11 和多个存储节点 22。图 1 仅是示例性说明，并不限定具体的组网方式，如：级联树形组网、环状组网都可以。只要主机 11 和存储节点 22 之间能够相互通信。

- 25 主机 11 可以包括任何计算设备，如服务器、台式计算机等等。用户可以通过主机 11 触发读写指令，向存储节点 22 发送写数据请求或读数据请求。

在本实施例中，主机 11 可以与任意一个存储节点 22 通信，并且任意两个存储节点 22 之间也可以通信。

如图 2 所示，主机 11 主要包括处理器(英文：processor)118、缓存（英文：cache）120、存储器(英文：memory)122、通信总线（简称总线）126 以及通信接口(英文：communication interface)128。处理器 118、缓存 120、存储器 122 以及通信接口 128 通过通信总线 126 完成相互间的通信。

通信接口 128，用于与存储节点 22 通信。

存储器 122，用于存放程序 124，存储器 122 可能包含高速 RAM 存储器，也可能还包括非易失性存储器（英文：non-volatile memory），例如至少一个磁盘存储器。可以理解的是，存储器 122 可以为随机存储器（英文：Random-Access Memory，RAM）、磁碟、硬盘、固态硬盘（英文：solid state disk，SSD）或者非易失性存储器等各种可以存储程序代码的机器可读介质。

程序 124 可以包括程序代码。

缓存 120（英文：cache）用于缓存从应用服务器 10 接收的数据或从存储节点 22 中读取的数据。缓存 120 可以是 RAM、ROM、闪存（英文：flash memory）或固态硬盘（Solid State Disk，SSD）等各种可以存储数据的机器可读介质，在此不做限定。

另外，存储器 122 和缓存 120 可以合设或者分开设置，本发明实施例对此不做限定。

处理器 118 可能是一个中央处理器（英文：central processing unit，CPU）用于生成并处理写数据请求等等。

存储节点 22 可以是磁盘阵列或者闪存阵列或者存储服务器。磁盘阵列是指包括至少一个控制器和多个磁盘的存储设备。闪存阵列是指包括至少一个控制器和多个固态硬盘（英文：solid state device，SSD）的存储设备。固态硬盘是以闪存（英文：flash memory）芯片为存储介质的存储器，又名固态硬盘驱动器（英文：solid state drive，SSD）。另外，存储节点 22 还可以是存储

服务器，所述存储服务器包含至少一个固态硬盘。

以闪存阵列为例，如图 3 所示，存储节点 22 包括控制器 221 和一个或多个固态硬盘 222。其中，控制器 221 至少包括接口 223、接口 224、处理器 225 和存储器 226。

- 5 接口 223，用于和主机 11 通信。接口 224，用于和固态硬盘 222 通信。处理器 225 可能是一个中央处理器（英文：central processing unit, CPU）。

处理器 225，用于接收来自主机 11 的写数据请求或者读数据请求、处理所述写数据请求或者读数据请求，处理器 225 还可以将写数据请求中的数据发送给固态硬盘 222。

- 10 存储器 226，用于存放程序。另外，存储器 226 还用于临时存储从主机 11 接收的写数据请求或从固态硬盘 222 读取的数据。控制器 221 接收主机发送的多个写数据请求时，可以将所述多个写数据请求暂时保存在存储器 226 中。当多个写数据请求的大小达到预设数据量时，将存储器 226 存储的多个写数据请求发送给校验计算节点。存储器 226 包含随机访问存储器(英文：
15 random-access memory, RAM)。可选地，存储器 226 还包括非易失性存储器（英文：non-volatile memory），例如至少一个磁存储器。可以理解的是，存储器 226 可以为随机存储器 (Random-Access Memory, RAM)、磁碟、硬盘、固态硬盘 (Solid State Disk, SSD) 或者非易失性存储器等各种可以存储程序代码的机器可读介质。

- 20 在本实施例中的多个存储节点中，用于存储主机 11 发送的写数据请求的存储节点被称为数据节点，用于计算多个写数据请求的校验数据的存储节点被称为校验计算节点，用于存储多个写数据请求的校验数据的存储节点被称为校验节点，用于计算元数据的校验数据的存储节点被称为元数据校验计算节点，用于进行系统垃圾回收的节点被称为垃圾回收节点。然而，这种划分并不是绝对的。例如，存储校验数据的校验节点也可以作为数据节点存储
25 写数据请求。

本申请提供了至少两种应用场景，在一种应用场景中，主机 11 具有计算多个写数据请求的校验数据的功能。举例来说，主机 11 根据写数据请求包括的数据待写入的数据节点的标识，将多个写数据请求划分为多个写数据请求集合，每个集合包括将要被写入相同的数据节点的写数据请求。当部分或者全部集合中的所有写数据请求的大小达到预设数据量时，主机 11 计算这些写数据请求集合的校验数据。

在另一种应用场景中，主机 11 不具有计算多个写数据请求的校验数据的功能，计算校验数据的操作是由存储节点中的校验计算节点来完成的。主机 11 根据每个写数据请求携带的数据待写入的数据节点的标识，将每个写数据请求发送给所述数据节点。每个数据节点接收主机 11 发送的多个写数据请求，当写数据请求集合（写数据请求集合包括多个写数据请求）的大小达到预设数据量时，数据节点将所述写数据请求集合发送给校验计算节点，由校验计算节点计算多个写数据请求集合的校验数据。

下面将分别讨论基于以上两种应用场景的数据校验方法。

图 4 是应用于第一种场景的数据校验方法的流程图，如图 4 所示，所述方法包括如下步骤。示例性的，步骤 S101-步骤 S104 可以由主机 11 的处理器 118 执行。

步骤 S101：主机 11 生成多个写数据请求，每个写数据请求包括数据、所述数据待写入的数据节点的标识以及所述数据待写入的逻辑地址。数据待写入的逻辑地址包括逻辑单元（英文：Logic Unit）的标识、逻辑块地址（英文：Logical Block Address, LBA）和长度（英文：length）。其中，逻辑单元的标识用于指示所述数据待写入的逻辑单元，逻辑块地址用于指示所述数据位于所述逻辑单元内的位置，长度表示所述数据的大小。所述数据待写入的数据节点的标识用于指示所述数据待写入的数据节点。所述数据待写入的数据节点是主机 11 根据所述数据或者所述数据待写入的逻辑地址从存储系统 10 中的多个数据节点中选择的。

具体的，首先，主机 11 预先收集存储系统 10 中的数据节点的信息。所述数据节点的信息包括存储系统 10 中包括的数据节点的个数，以及每个数据节点的标识。数据节点的标识用于唯一识别所述数据节点。例如，存储系统 10 包含 5 个数据节点，每个数据节点的标识分别是 A、B、C、D 和 E。

- 5 主机 11 可以向每个数据节点发送查询请求以获得其标识，也可以由各个数据节点主动向主机 11 上报其标识，还可以在存储系统 10 中指定一个主节点，由所述主节点汇总各个数据节点的标识之后，向主机 11 上报所述各个数据节点的标识。

- 其次，主机 11 确定数据待写入的数据节点。一种实现方式是主机 11 根据所述数据待写入的逻辑地址确定所述数据待写入的数据节点。例如：主机 11 将每个写数据请求包括的数据待写入的数据节点的地址作为输入项，利用预先设定的哈希算法得到哈希值，所述哈希值唯一对应一个数据节点的标识。另外，主机 11 还可以利用散列算法或者取余等方式确定数据节点的标识。本实施例并不对确定数据节点的标识的算法进行任何限定，只要能根据
- 15 所述逻辑地址唯一确定一个数据节点的标识即可。另一种实现方式是主机 11 根据所述数据确定所述数据待写入的数据节点。例如：主机 11 将数据作为输入项，利用预先设定的哈希算法得到哈希值，所述哈希值唯一对应一个数据节点的标识。同样的，主机 11 还可以利用散列算法或者取余等方式确定数据节点的标识。再一种实现方式是，主机 11 根据触发所述写数据请求的用户对多个写数据请求进行分类，为相同用户触发的写数据请求选择同一个数据节点。或者，主机 11 还可以根据数据的类型对这些写数据请求进行分类，为相同类型的数据选择同一个数据节点。
- 20

- 主机 11 在为数据选择了数据节点之后，将所述数据节点的标识写入携带所述数据的写数据请求中。所述数据节点的标识可以位于所述写数据请求的头部，也可以是所述写数据请求的载荷（英文：payload）。
- 25

用户在主机 11 中触发多个写数据请求时，向主机 11 发送指令要求将这

些写数据请求写入同一个数据节点。主机 11 可以根据用户的需求，为这些写数据请求选择一个数据节点，将这个数据节点携带在每个写数据请求。

5 由于每个写数据请求中的数据节点的标识是主机 11 根据所述写数据请求包括的数据或者数据待写入的逻辑地址分配的，因此具有同样的标识的写数据请求属于相同类别的写数据请求，这些写数据请求所携带的数据被同时读取的可能性较大。

步骤 S102：主机根据写数据请求中包括的数据节点的标识，将所述多个写数据请求划分为多个写数据请求集合，每个集合包括具有相同的数据节点
10 点的标识的多个写数据请求。

需要说明的是，在实际实现时，并不需要将所述多个写数据请求存储在不同的地方，这里的“划分”仅仅是逻辑上的，以不同的数据节点的标识对所述多个写数据请求进行分类。例如，一个写数据请求集合中的多个写数据请求均包含数据节点 A 的标识，另一个写数据请求集合中的多个写数据请求均包含数据节点 B 的标识，再一个写数据请求集合中的多个写数据
15 请求均包含数据节点 C 的标识。

步骤 S103：当设定数量个写数据请求集合中的每个写数据请求集合中的所有写数据请求的大小达到预设数据量时，主机 11 计算所述设定数量个写数据请求集合的校验数据。

20 通常情况下，存储系统 10 包含若干个数据节点，相应地，主机 11 中的写数据请求集合也有若干个。随着生成的写数据请求的不断增多，每个写数据请求集合的大小也不断累积。主机 11 需要从大小达到预设数据量的写数据请求集合中选择设定数量个写数据请求集合，计算所述设定数量个写数据请求集合的校验数据。所述预设数据量是预先设定的，例如 16KB。所述设
25 定数量是由存储系统 10 预设设定的校验模式确定的。

预先设定的校验模式包括 5+1 模式，或者 6+2 模式等等。5+1 模式是指

根据 5 个写数据请求集合生成一个校验数据；6+2 模式是指根据 6 个写数据请求集合生成 2 个校验数据。以 5+1 模式为例，主机 11 从凑满预设数据量的多个写数据请求集合中选择 5 个写数据请求集合，再计算这 5 个写数据请求集合的校验数据。这 5 个写数据请求集合和校验数据就组成了一个分条(英文：stripe)。大小达到预设数据量的写数据请求集合就是所述分条的一个数据单元。

步骤 S104，主机 11 将所述设定数量个写数据请求集合中的每个写数据请求集合发送给写数据请求集合包含的标识所指示的数据节点。

由前面的讨论可知，由于每个写数据请求集合中的写数据请求都包含相同的数据节点的标识，，因此主机可以根据所述标识将每个写数据请求集合发送给所述标识所指示的数据节点。

步骤 S105，主机 11 从存储系统 10 选择一个存储节点用于存储所述校验数据，将所述校验数据发送给所述选择出的存储节点，所述用于存储校验数据的存储节点与组成这个分条的每个写数据请求集合所在的数据节点不相同。为了描述方便，将存储校验数据的存储节点称为校验节点。

主机 11 可以从剩余的存储节点（排除用于保存写数据请求集合的数据节点）中选择出一个或多个，用于保存校验数据。通常情况下，对于一个分条而言，写数据请求集合和校验数据不能保存在同一个存储节点中。这是为了防止一个存储节点发生故障时，写数据请求集合和校验数据同时丢失。然而，在实际应用中，主机 11 中通常可以构建多个分条。虽然，对于一个分条而言，选择出的用于保存校验数据的存储节点不能是保存这个分条的写数据请求集合的存储节点，但可以是保存有其他分条的写数据请求集合的存储节点。本实施例并没有限定某个存储节点专门用于存储写数据请求集合或者校验数据。

另外，主机 11 给每个分条分配分条标识，不同的分条具有不同的分条标识。并且，主机给每个分条包含的各个写数据请求集合以及校验数据分配

标识, 将写数据请求集合的标识发送给所述写数据请求集合所在的数据节点。对于每个分条, 主机 11 记录分条标识与所述分条包含的每个写数据请求集合的标识之间的对应关系以及所述分条标识与校验数据的标识之间的对应关系。

- 5 按照图 4 所示的数据校验方法, 主机 11 按照写数据请求包含的数据节点的标识, 将多个写数据请求划分为多个写数据请求集合。当设定数量个写数据请求集合中所有写数据请求的大小达到预设数据量时, 主机 11 计算所述设定数量个写数据请求集合的校验数据, 并将校验数据发送给校验节点存储, 由此保证了数据的可靠性。并且, 由于每个写数据请求集合包含待写入
- 10 相同的数据节点的多个写数据请求, 而每个写数据请求待写入的数据节点是所述主机根据所述写数据请求中的数据或者所述数据待写入的逻辑地址选择的。因此, 每个写数据请求集合中保存的是相同类别的写数据请求。当主机 11 计算校验数据之后, 将每个写数据请求集合发送给所述写数据请求集合包含的标识所指示的数据节点, 相同类别的写数据请求就被存储在了同一
- 15 个数据节点中。由于相同类别的写数据请求中的数据被同时读取的可能性较大, 那么在这些数据被读取时可以在一个数据节点内读取, 不用跨节点执行读取操作, 提高了数据读取的效率。

- 应用于第一种场景的数据校验方法还有另外一种实现方式, 与图 4 所示的实现方式不同之处在于, 在这种实现方式中, 是由校验计算节点按照写数
- 20 据请求包含的数据节点的标识将接收的多个写数据请求划分为多个写数据请求集合, 当设定数量个写数据请求集合达到预设数据量时, 计算校验数据。而这些操作在图 4 所示的实现方式中是由主机 11 完成的。具体的, 如图 5 所示, 这种数据校验方法可以通过以下步骤实现。

- 步骤 S401: 该步骤与图 4 所示的步骤 S101 类似, 这里不再赘述。
- 25 步骤 S402: 主机 11 将生成的写数据请求发送给校验计算节点。
- 步骤 S403: 所述校验计算节点根据写数据请求中包括的数据节点的标

识，将所述多个写数据请求划分为多个写数据请求集合，每个集合包括具有相同的数据节点的标识的多个写数据请求。与图 4 不同之处在于该步骤的执行主体是校验计算节点，其余部分与图 4 所示的步骤 S102 类似，这里不再赘述。

- 5 步骤 S404：当设定数量个写数据请求集合中的每个写数据请求集合中的所有写数据请求的大小达到预设数据量时，所述校验计算节点计算所述设定数量个写数据请求集合的校验数据。与图 4 不同之处在于该步骤的执行主体是校验计算节点，其余部分与图 4 所示的步骤 S103 类似，这里不再赘述。

- 10 步骤 S405：所述校验计算节点将所述设定数量个写数据请求集合中的每个写数据请求集合发送给写数据请求集合包含的标识所指示的数据节点。与图 4 不同之处在于该步骤的执行主体是校验计算节点，其余部分与图 4 所示的步骤 S104 类似，这里不再赘述。

- 15 所述校验计算节点在计算出校验数据之后，可以直接将所述校验数据保存在本地，不用转发给其他存储节点存储。如果按照预先设定的校验模式，计算得出的检验数据不止一个，那么所述校验计算节点可以从存储系统 10 选择一个存储节点用于存储另一个校验数据。选择方式与与图 4 所示的步骤 S105 类似，这里不再赘述。图 6 是应用于第二种场景的数据校验方法的流程图，如图 6 所示，所述方法包括如下步骤：

- 20 步骤 S201：主机 11 生成多个写数据请求，每个写数据请求包括数据、所述数据待写入的数据节点的标识以及所述数据待写入的逻辑地址。步骤 S201 的内容与图 4 所示的方法中的步骤 S101 类似，这里不再赘述。

步骤 S202：主机 11 将每个写数据请求发送给所述写数据请求包含的数据待写入的数据节点的标识所指示的数据节点。

- 25 步骤 S203：数据节点接收主机 11 发送的多个写数据请求，将所述写入写数据请求写入存储器 226 中。具体的，该步骤可以由控制器 221 中的处理器 225 执行。

步骤 S204: 当数据节点的存储器 226 中保存的所有写数据请求的大小达到预设数据量时, 数据节点将写数据请求集合发送给校验计算节点。所述写数据请求集合包括多个写数据请求, 并且所述多个写数据请求的大小达到预设数据量。

5 具体的, 该步骤可以由控制器 221 中的处理器 225 执行。每当数据节点接收主机 11 发送的写数据请求, 就会将该写数据请求写入存储器 226。因此存储器 226 会逐渐积累写数据请求, 当积累的写数据请求的大小达到预设数据量时, 该数据节点将这些积累的写数据请求作为一个集合发送给校验计算节点。所述预设数据量是所述存储系统 10 预先设定的, 例如 16KB。

10 对于任意一个数据节点而言, 只要其存储器 226 中所有写数据请求的大小达到所述预设数据量, 就会将存储器 226 中所有写数据请求发送给校验计算节点, 不会考虑其他数据节点的存储器中累积的写数据请求是否达到所述预设数据量。在实际应用中, 数据节点内部可能对积累写数据请求有时限的要求, 那么如果当预设时限到达时, 存储器 226 中所有写数据请求的大小尚未到达所述预设数据量, 那么可以用 0 或者其他特殊标记的数据补足。

 上面的描述是针对一个数据节点而言的, 那么对于存储系统 10 中的多个数据节点来说, 每个数据节点 22 都会先在本本地凑满达到预设数据量的写数据请求集合, 然后这些写数据请求集合发送给校验计算节点。校验计算节点根据接收的多个写数据请求集合计算校验数据, 从而组成一个分条。

20 步骤 S205: 校验计算节点接收多个写数据请求集合, 从所述多个写数据请求集合中选择设定数量个写数据请求集合, 所述设定数量个写数据请求集合是所述多个写数据请求集合的子集, 并且所述设定数量个写数据请求集合包括不同的数据节点的标识, 计算所述设定数量个写数据请求集合的校验数据。

25 以存储节点 C 为校验计算节点为例, 存储节点 C 可以与存储系统 10 中的任意一个存储节点 22 通信。存储节点 C 接收多个写数据请求集合之后,

按照预设设定的校验模式从所述多个写数据请求集合中选择设定数量个写数据请求集合，计算所述设定数量个写数据请求集合的校验数据。以 5+1 模式为例，当存储节点 C 从接收的多个写数据请求集合中选择 5 个来自不同数据节点的写数据请求集合，计算这 5 个写数据请求集合的校验数据。在某些情况下，存储节点 C 本地保存的写数据请求集合也可以参与校验数据的计算，则只需要接收其余 4 个数据节点发送的写数据请求集合即可。

当校验计算节点接收一个写数据请求集合时，保存所述写数据请求集合，为所述写数据请求集合分配标识，然后将所述写数据请求集合的标识写入链表或者其他数据结构中。所述写数据请求集合的标识可以与所述写数据请求集合中的写数据请求包括的数据节点的标识相同，也可以与所述写数据请求集合中的写数据请求包括的数据节点的标识不同。当所述写数据请求集合的标识与所述写数据请求集合中的写数据请求包括的数据节点的标识不同时，校验计算节点保存所述写数据请求集合的标识与数据节点的标识之间的对应关系。在一种实现方式中，所述写数据请求集合的标识是由校验计算节点分配，校验计算节点分配标识以后再将所述写数据请求集合的标识发送给所述写数据请求集合所在的数据节点，所述数据节点保存所述写数据请求集合的标识。在另一种实现方式中，所述写数据请求集合的标识是由所述写数据请求集合所在的数据节点分配的，数据节点分配标识以后再将所述写数据请求集合的标识发送给所述校验计算节点，所述校验计算节点保存所述写数据请求集合的标识。

为了保证数据的可靠性，校验计算节点在选择一定数目的写数据请求集合组成分条时，需保证组成同一个分条的多个写数据请求集合均来自不同的数据节点。以 5+1 模式为例，校验计算节点需要从所述链表中选择 5 个写数据请求集合。由于校验计算节点在一定时间内可能会接收多个写数据请求集合，其中某些写数据请求集合可能是来自同一个数据节点。因此，校验计算节点还需要保证这 5 个写数据请求集合来自不同的数据节点。具体的，校验

计算节点可以根据保存的写数据请求集合的标识与数据节点的标识之间的对应关系，或者直接根据写数据请求集合的标识从链表中选择 5 个写数据请求集合。需要说明的是，校验计算节点在选择组成分条的写数据请求集合时，并不需要考虑各个写数据请求集合进入所述链表的先后顺序，只需保证组成

5 该分条的多个写数据请求集合来自不同的数据节点。

步骤 S206：校验计算节点将校验数据发送给校验节点，所述校验节点不能和保存写数据请求集合的数据节点相同。

具体的，校验计算节点可以从剩余的存储节点（排除已经保存有写数据请求集合的数据节点）中选择一个或多个，用于保存校验数据。用于保存
10 校验数据的存储节点，在本实施例中称为校验节点。校验节点和所述设定数量个写数据请求集合所在的任意一个数据节点不能重复，这是为了防止一个存储节点发生故障时，一个分条所包含的写数据请求集合和校验数据同时丢失。然而，虽然选择出的用于保存校验数据的存储节点不能是保存是当前分条的写数据请求集合的数据节点，但可以是保存有其他分条的写数据请求集合的数据节点。本实施例并没有限定某个存储节点专门用于存储写数据请求
15 集合或者校验数据。

另外，校验计算节点给每个分条分配分条标识，不同的分条具有不同的分条标识。并且，主机给每个分条包含的各个写数据请求集合以及校验数据分配标识，将写数据请求集合的标识发送给所述写数据请求集合所在的数据
20 节点。或者，每个数据节点为保存的写数据集合分配标识，将所述写数据请求集合的标识发送给校验计算节点。对于每个分条，校验计算节点记录分条标识与所述分条包含的每个写数据请求集合的标识之间的对应关系以及所述分条标识与校验数据的标识之间的对应关系。

按照图 6 所示的数据校验方法，每个数据节点在累积的所有写数据请求
25 的大小达到预设数据量时，将这些写数据请求作为一个写数据请求集合发送给校验计算节点。校验计算节点从接收的多个写数据请求集合中选择设定数

量个写数据请求集合计算所述设定数量个写数据请求集合的校验数据，并将校验数据发送给校验节点存储，由此保证了数据的可靠性。并且，由于每个写数据请求集合包含具有相同的数据节点的标识的多个写数据请求，而每个写数据请求待写入的数据节点是所述主机根据所述写数据请求中的数据或者者所述数据待写入的逻辑地址选择的。因此，每个写数据请求集合中保存的是相同类别的写数据请求。由于相同类别的写数据请求中的数据被同时读取的可能性较大，那么在这些数据被读取时可以在一个数据节点内读取，不用跨节点执行读取操作，提高了数据读取的效率。

图 4 或图 5 或图 6 所示的数据校验方法实现了将相同类别的写数据请求写入同一个数据节点。在写数据请求写入数据节点之后，数据节点还需要创建并保存元数据。

元数据包括写数据请求集合的标识与每个写数据请求中的数据待写入的逻辑地址之间的对应关系。元数据还包括每个写数据请求在所述写数据请求集合中的位置。本实施例将每个写数据请求位于所述写数据请求集合中的位置称为内部偏移量。示例性的，写数据请求集合与其包含的各个写数据请求的数据待写入的逻辑地址之间的对应关系如表 1 所示：

逻辑地址	写数据请求集合的标识	内部偏移量
0x1+0x100+4096	1	0x1000
0x2+0x400+4096	1	0x2000
0x3+0x800+4096	1	0x3000

表 1

如上表所示，标识为“1”的写数据请求集合由三个写数据请求组成，每个写数据请求除了包括数据之外还包括数据待写入的逻辑地址，另外，每个写数据请求在写数据请求集合内部都有自己对应的位置。以第一条写数据请求为例，所述写数据请求的逻辑地址包括卷 ID、LBA 和 length，其中卷 ID 为 0x1，LBA 为 0x100，length 为 4096。所述写数据请求属于标识为“1”

的写数据请求集合，并且在所述写数据请求集合中的位置为 0x1000。

如果由主机将多个写数据请求划分为多个数据单元，计算多个数据单元的校验数据之后，把各个数据单元以及校验数据发送给系统中的多个存储节点存储。那么，元数据（写数据请求与数据单元之间的对应关系）是保存在主机中的。当后续在某个存储节点内对数据单元进行垃圾回收等操作时，需要给主机发送消息修改元数据，操作复杂并且耗费带宽。

然而，在本实施例中，由于相同类别的写数据请求作为一个写数据请求集合保存在一个数据节点，那么与这个写数据请求集合相关的元数据也保存在数据节点本地。后续如果对写数据请求集合进行垃圾回收等操作，可以直接在本地对元数据进行修改。

元数据可以保存在数据节点的存储器 226 中。为了保证元数据的可靠性，也需要计算并保存各个数据节点中的元数据的校验数据。具体的，随着数据节点接收的写数据请求增多，存储器 226 中保存的元数据也会增多。当存储器 226 中累积的元数据达到预设数据量时，数据节点将元数据集合发送给元数据校验计算节点。元数据集合是指大小达到预设数据量的元数据。所述预设数据量可以是所述存储系统 10 预先设定的，例如 16KB。元数据校验计算节点是指存储节点中用于计算元数据集合的校验数据的存储节点。在实际实现时可以和校验计算节点相同。另外，在本实施例中，数据节点 22 除了将元数据集合发送给元数据校验计算节点，还需要将数据节点的标识发送给所述元数据校验计算节点。所述元数据集合和标识可以封装在一个消息中发送，也可以分别发送。

元数据校验计算节点接收每个数据节点发送的元数据集合和数据节点的标识，为每个元数据集合分配标识，并保存元数据集合的标识与数据节点的标识之间的对应关系。为了保证元数据的可靠性，元数据校验计算节点计算元数据集合的校验数据。具体的，元数据校验计算节点根据所述元数据集合的标识与数据节点的标识之间的对应关系，从多个元数据集合中选择设定

数量个元数据集合，计算选择出的设定数量个元数据集合的校验数据。所述选择出的元数据集合需对应不同的数据节点的标识。

与前面的描述类似，所谓设定数量是由存储系统 10 预先设定的校验模式决定的。需要说明的是，元数据集合的校验模式可以与写数据请求集合的
5 校验模式相同，也可以不同。

元数据校验计算节点可以在存储系统 10 选择一个存储节点作为元数据校验节点。将所述元数据集合的校验数据发送给所述元数据校验节点。所述选择出的元数据校验节点不同于用于存储所述元数据集合的数据节点。然而，所述选择出的元数据校验节点可以是保存其他分条的元数据集合的数据
10 节点。本实施例并没有限定某个数据节点专门用于存储元数据集合或者元数据集合的校验数据。

对于每个数据节点来说，当存储器 226 中保存的数据总量达到一定阈值时，则数据节点需要将存储器 226 中的数据写入固态硬盘 222 中。此时，数据节点中的控制器 221 可以以每个写数据请求集合为粒度为这个集合分配写入固态硬盘的逻辑地址（本实施例中称为硬盘逻辑地址），并保存所述写数据
15 请求集合的标识与分配的硬盘逻辑地址之间的对应关系。固态硬盘 222 接收控制器 221 发送到写数据请求集合以及所述为所述集合分配的写入固态硬盘的逻辑地址，将所述写数据请求集合写入一个或多个块（英文：block）中。并且，固态硬盘 222 保存所述为所述集合分配的写入固态硬盘的逻辑地址与
20 实际地址之间的对应关系。实际地址是指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，只对固态硬盘可见的地址。

对于每个数据节点保存的写数据请求集合，经过一段时间之后，写数据请求集合中的部分数据可能成为无效数据。数据的有效性是以所述数据是否
25 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）。以表 1 为例，当标识为 1 的写数据请求集合所在的数据节点接收到第四条写数据请求，并且第四条写数据请求中的数据待写入的逻辑地址与第一条写数据请求中的数据待写入的逻辑地址相同，那么说明第四条写数据请求是用于覆盖所述第一条写数据请求的。由于固态硬盘的写入特性，第四条写数据请求中的数据（新数据）不会直接覆盖第一条写数据请求的数据（旧数据），而是在固态硬盘上分配一块空白的块，写入所述新数据。从而，第一条写数据请求中的数据就成了无效数据。也意味着，标识为 1 的写数据请求集合中的部分数据成为无效数据。数据节点可以用位图记录每个写数据请求集合包含的无效数据的信息，所述无效数据的信息包括无效数据待写入的逻辑地址以及无效数据的数据量。例如，位图的每个“位”对应大小为 1KB 的数据待写入的逻辑地址，当“位”为 1 时，代表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。

随着无效数据的增多，数据节点需要进行系统垃圾回收操作。系统垃圾回收以分条为单位执行，由于分条包含的写数据请求集合分布在数据节点中。因此，在数据节点内部以写数据请求集合为单位执行系统垃圾回收。存储系统还包括垃圾回收节点，垃圾回收节点是存储节点 22 中用于执行系统垃圾回收操作的节点。图 7 是系统垃圾回收方法的流程示意图，如图 7 所示，可以包括如下步骤：

20 S301，垃圾回收节点根据每个分条包含的写数据请求集合的位图，从多个分条选择待回收的分条。

在第一种应用场景中，主机 11 将分条标识与所述分条标识与所述分条包含的每个写数据请求集合的标识之间的对应关系发送给垃圾回收节点。在第二种应用场景中，校验计算节点将分条标识与所述分条标识与所述分条包含的每个写数据请求集合的标识之间的对应关系发送给垃圾回收节点。另
25 外，无论是第一种应用场景还是第二种应用场景，数据节点都需要将写数据

请求集合的位图发送给垃圾回收节点。

垃圾回收节点需要从多个分条中选择一个待回收的分条。为了使系统垃圾回收操作的回收效率最高，垃圾回收节点通常选择包含无效数据最多的分条作为待回收的分条，因此，需要统计每个分条包含的无效数据的数据量。

- 5 具体的，垃圾回收节点可以根据分条标识与所述分条包含的每个写数据请求集合的标识之间的对应关系确定该分条包含的所有的写数据请求集合，再根据每个写数据请求集合的位图确定每个写数据请求集合所包含的无效数据的数据量，从而确认所述分条所包含的无效数据量。由此，垃圾回收节点可以将包含无效数据最多的分条作为待回收的分条。此外，本实施例也可以根据其他条件选择待回收的分条，例如，较长时间内未进行系统垃圾回收操作的分条，或者按照预设的顺序依次对各个分条进行系统垃圾回收。
- 10

S302，垃圾回收节点向每个写数据请求集合所在的数据节点发送垃圾回收通知消息。

- 如前面所述，垃圾回收节点可以根据分条标识与所述分条标识与所述分条包含的每个写数据请求集合的标识之间的对应关系确定每个分条所包含的写数据请求集合的标识。然后，根据写数据请求集合的标识与数据节点的标识之间的对应关系，确定每个写数据请求集合所在的数据节点（如果写数据请求集合的标识与数据节点的标识一致，则可以直接根据写数据请求集合的标识确定每个写数据请求集合所在的数据节点），从而向每个写数据请求集合所在的数据节点发送垃圾回收通知消息，所述消息用于通知数据节点对所述写数据请求集合进行系统垃圾回收操作。所述消息包括所述写数据请求集合的标识。
- 15
- 20

- S303，每个数据节点接收所述垃圾回收通知消息，根据所述消息中携带的写数据请求集合的标识确定待回收的写数据请求集合。然后，每个数据节点再根据所述待回收的写数据请求集合的位图确定所述待回收的写数据请求集合包含的有效数据和无效数据。例如，表1的第一条写数据请求的数据
- 25

变成了无效数据。第二条写数据请求的数据、第三条写数据请求的数据仍然是有效数据。

S304, 每个数据节点将所述待回收的写数据请求集合中的有效数据迁移到新的写数据请求集合中, 并回收所述写数据请求集合。

- 5 将有效数据迁移到新的写数据请求集合是指将所述有效数据所在的写数据请求拼凑在新的写数据请求集合中。例如, 将表 1 中的第二条写数据请求以及第三条写数据请求从固态硬盘 222 中读入存储器 226 (如果第二条写数据请求或者第三条写数据请求尚且保存在存储器 226 中, 则不需要执行此操作), 待存储器 226 中存储的所有写数据请求的大小重新达到预设数据量
- 10 时, 就凑满一个新的写数据请求集合。具体的, 数据节点再根据所述待回收的写数据请求集合的位图确定所述写数据请求集合中包含的有效数据待写入的逻辑地址 (这里的逻辑地址是指数据待写入的逻辑地址), 删除所述有效数据待写入的逻辑地址与所述写数据请求集合的标识之间的对应关系, 并且保存所述有效数据待写入的逻辑地址与新的写数据请求集合的标识之间
- 15 的对应关系。由于这些元数据保存在数据节点本地, 因此数据节点在执行系统垃圾回收之后可以直接在本地修改元数据, 无需跨节点操作。然后, 数据节点将所述待回收的写数据请求集合的硬盘逻辑地址发送给固态硬盘, 由固态硬盘将与所述硬盘逻辑地址对应的块标记为无效。当后续固态硬盘进行内部的垃圾回收时可以直接擦除这些块, 无需再次进行有效数据的复制, 减少
- 20 了固态硬盘内部写放大的次数。另外, 数据节点还需要删除所述待回收的写数据请求集合的标识与分配的硬盘逻辑地址之间的对应关系。当一个分条所包含的所有写数据请求集合都被回收了, 那么该分条所包含的校验数据也没有存在的意义了, 垃圾回收节点可以通知校验节点删除所述校验数据。

- 本实施例提供的系统垃圾回收是由每个数据节点将有效数据拼凑在本
- 25 地的其他写数据请求集合中, 因此所述数据节点独立完成系统垃圾回收, 无需和其他数据节点进行数据交互, 节省了数据节点之间的带宽。

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。

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，取决于技术方案的具体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，包括主机、校验节点和多个数据节点，每个数据节点具有唯一的标识，

5 所述主机，用于根据写数据请求中包括的数据节点的标识，将多个写数据请求划分为多个写数据请求集合，每个写数据请求集合包括具有相同的数据节点的标识的多个写数据请求，每个写数据请求包括数据、所述数据待写入的逻辑地址以及所述数据待写入的数据节点的标识，所述数据待写入的数据节点是所述主机根据所述数据或所述逻辑地址选择的；

10 当设定数量个写数据请求集合中的每个写数据请求集合中的所有写数据请求的大小达到预设数据量时，计算所述设定数量个写数据请求集合的校验数据，所述设定数量个写数据请求集合是所述多个写数据请求集合的子集；

 将所述设定数量个写数据请求集合中的每个写数据请求集合发送给所述写数据请求集合包括的数据节点的标识所指示的数据节点；
15 将所述校验数据发送给所述校验节点。

2、根据权利要求 1 所述的存储系统，其特征在于，

 所述主机，还用于为所述设定数量个写数据请求集合中的每个写数据请求集合分配标识，并且将每个写数据请求集合的标识发送给该写数据请求集合包含的数据节点的标识所指示的数据节点；
20 求集合分配标识，并且将每个写数据请求集合的标识发送给该写数据请求集合包含的数据节点的标识所指示的数据节点；

 所述数据节点，用于接收写数据请求集合以及所述写数据请求集合的标识；创建元数据，所述元数据包括所述写数据请求集合的标识与每个写数据请求中的数据待写入的逻辑地址之间的对应关系，以及每个写数据请求中的数据待写入的逻辑地址与内部偏移量之间的对应关系。
25 数据待写入的逻辑地址与内部偏移量之间的对应关系。

3、根据权利要求 2 所述的存储系统，其特征在于，所述存储系统还包

括元数据校验计算节点以及元数据校验节点；

所述数据节点，还用于在确定累积的元数据达到所述预设数据量时，将元数据集合以及所述数据节点的标识发送给所述元数据校验计算节点，所述元数据集合包括所述累积的达到所述预设数据量的元数据；

- 5 所述元数据校验计算节点，用于接收多个数据节点中每个数据节点发送的元数据集合以及所述数据节点的标识，保存每个元数据集合与数据节点的标识之间的对应关系，根据所述对应关系从接收的多个元数据集合中选择设定数量个元数据集合，所述选择出的设定数据量个元数据集合对应不同的数据节点的标识；计算所述选择出的设定数量个元数据集合的校验数据；将所
10 述设定数量个元数据集合的校验数据发送给所述元数据校验节点，所述元数据校验节点不同于存储所述设定数量个元数据集合中的每个元数据集合所在的数据节点。

- 4、根据权利要求2所述的存储系统，其特征在于，所述存储系统还包
15 括垃圾回收节点；

所述主机，还用于为分条分配标识，所述分条包括所述设定数量个写数据请求集合以及所述设定数量个写数据请求集合的校验数据；将所述分条标识与所述分条包括的每个写数据请求集合的标识之间的对应关系，发送给垃圾回收节点；

- 20 所述数据节点，还用于将保存的写数据请求集合的位图发送给所述垃圾回收节点，所述位图用于指示所述写数据请求集合的无效数据的数据量；

- 所述垃圾回收节点，用于根据所述分条标识与所述分条包括的每个写数据请求集合的标识之间的对应关系，以及每个写数据请求集合的位图确定所述分条包含的无效数据的数据量大于其他任意一个分条包含的无效数据的
25 数据量，并且在确定所述分条包含的无效数据的数据量大于其他任意一个分条包含的无效数据的数据量时，向所述分条包含的每个写数据请求集合所在

的数据节点分别发送垃圾回收通知消息，每个垃圾回收通知消息包括该写数据请求集合的标识。

5、根据权利要求4所述的存储系统，其特征在于，

- 5 所述数据节点，还用于在接收所述垃圾回收通知消息之后，根据所述写数据请求集合的标识以及保存的所述写数据请求集合的位图，对所述写数据请求集合执行系统垃圾回收操作。

- 6、根据权利要求1-5任一所述的存储系统，其特征在于，所述设定数量是根据预先设定的校验模式确定的。
- 10

7、一种数据校验方法，其特征在于，所述方法用于存储系统中，所述存储系统包括主机、校验节点和多个数据节点，每个数据节点具有唯一的标识，

- 15 所述主机根据写数据请求中包括对数据节点的标识，将多个写数据请求划分为多个写数据请求集合，每个写数据请求集合包括具有相同的数据节点的标识的多个写数据请求，每个写数据请求包括数据、所述数据待写入的逻辑地址以及所述数据待写入的数据节点的标识，所述数据待写入的数据节点是所述主机根据所述数据或所述逻辑地址选择的；

- 20 当设定数量个写数据请求集合中的每个写数据请求集合中的所有写数据请求的大小达到预设数据量时，所述主机计算所述设定数量个写数据请求集合的校验数据，所述设定数量个写数据请求集合是所述多个写数据请求集合的子集；

- 25 所述主机将所述设定数量个写数据请求集合中的每个写数据请求集合发送给所述写数据请求集合包括的数据节点的标识所指示的数据节点；

所述主机将所述校验数据发送给所述校验节点。

8、根据权利要求7所述的方法，其特征在于，还包括：

所述主机为所述设定数量个写数据请求集合中的每个写数据请求集合分配标识，将每个写数据请求集合的标识发送给该写数据请求集合包含的数据节点的标识所述指示的数据节点；

5 所述数据节点接收写数据请求集合以及所述写数据请求集合的标识；创建元数据，所述元数据包括所述写数据请求集合的标识与每个写数据请求中的数据待写入的逻辑地址之间的对应关系，以及每个写数据请求中的数据待写入的逻辑地址与内部偏移量之间的对应关系。

10 9、根据权利要求8所述的方法，其特征在于，所述存储系统还包括元数据校验计算节点以及元数据校验节点，所述方法还包括：

所述数据节点在确定累积的元数据达到所述预设数据量时，将元数据集合以及所述数据节点的标识发送给所述元数据校验计算节点，所述元数据集合包括所述累积的达到所述预设数据量的元数据；

15 所述元数据校验计算节点接收多个数据节点中每个数据节点发送的元数据集合以及所述数据节点的标识；

所述元数据校验计算节点保存每个元数据集合与数据节点的标识之间的对应关系；

20 所述元数据校验计算节点根据所述对应关系从接收的多个元数据集合中选择设定数量个元数据集合，所述选择出的设定数据量个元数据集合对应不同的数据节点的标识；

所述元数据校验计算节点计算所述选择出的设定数量个元数据集合的校验数据，将所述设定数量个元数据集合的校验数据发送给所述元数据校验节点，所述元数据校验节点不同于存储所述设定数量个元数据集合中的每个
25 元数据集合所在的数据节点。

10、根据权利要求 8 所述的方法，其特征在于，所述存储系统还包括垃圾回收节点，所述方法还包括：

所述主机为分条分配标识，所述分条包括所述设定数量个写数据请求集合以及所述设定数量个写数据请求集合的校验数据；

5 所述主机将所述分条标识与所述分条包括的每个写数据请求集合的标识之间的对应关系，发送给垃圾回收节点；

所述数据节点将保存的写数据请求集合的位图发送给所述垃圾回收节点，所述位图用于指示所述写数据请求集合的无效数据的数据量；

10 所述垃圾回收节点根据所述分条标识与所述分条包括的每个写数据请求集合的标识之间的对应关系，以及每个写数据请求集合的位图确定所述分条包含的无效数据的数据量大于其他任意一个分条包含的无效数据的数据量；

15 所述垃圾回收节点在确定所述分条包含的无效数据的数据量大于其他任意一个分条包含的无效数据的数据量时，向所述分条包含的每个写数据请求集合所在的数据节点分别发送垃圾回收通知消息，每个垃圾回收通知消息包括该写数据请求集合的标识。

11、根据权利要求 10 所述的方法，其特征在于，还包括：

20 所述数据节点在接收所述垃圾回收通知消息之后，根据所述写数据请求集合的标识以及保存的所述写数据请求集合的位图，对所述写数据请求集合执行系统垃圾回收操作。

25 12、根据权利要求 7-11 任一所述的方法，其特征在于，所述设定数量是根据预先设定的校验模式确定的。

13、一种主机，其特征在于，包括通信接口和处理器；

所述通信接口，用于与校验节点、以及多个数据节点通信，其中，每个

数据节点具有唯一的标识；

所述处理器，用于根据写数据请求中包括的数据节点的标识，将多个写数据请求划分为多个写数据请求集合，每个写数据请求集合包括具有相同的数据节点的标识的多个写数据请求，每个写数据请求包括数据、所述数据待写入的逻辑地址以及所述数据待写入的数据节点的标识，所述数据待写入的数据节点是所述主机根据所述数据或所述逻辑地址选择的；

当设定数量个写数据请求集合中的每个写数据请求集合中的所有写数据请求的大小达到预设数据量时，计算所述设定数量个写数据请求集合的校验数据，所述设定数量个写数据请求集合是所述多个写数据请求集合的子集；

将所述设定数量个写数据请求集合中的每个写数据请求集合发送给所述写数据请求集合包括的数据节点的标识所指示的数据节点；

将所述校验数据发送给所述校验节点。

14、一种存储系统，其特征在于，所述存储系统包括校验节点、校验计算节点和多个数据节点，每个数据节点具有唯一的标识，

每个数据节点，用于将写数据请求集合发送给所述校验计算节点，所述写数据请求集合包括一个或多个写数据请求，每个写数据请求包括数据以及所述数据待写入的数据节点的标识，所述写数据请求集合的大小等于预设数据量；

所述校验计算节点，用于接收多个写数据请求集合；从所述多个写数据请求集合中选择设定数量个写数据请求集合，所述设定数量个写数据请求集合是所述多个写数据请求集合的子集，并且所述设定数量个写数据请求集合包括不同的数据节点的标识；计算所述设定数量个写数据请求集合的校验数据；将所述校验数据发送给所述校验节点，所述校验节点不同于所述设定数量个写数据请求集合中的每个写数据请求集合所在的数据节点。

15、根据权利要求 14 所述的存储系统，其特征在于，每个写数据请求还包括所述数据待写入的逻辑地址，所述数据待写入的数据节点是所述存储系统中的主机根据所述数据或者所述数据待写入的逻辑地址选择的；

每个数据节点，还用于接收所述主机发送的多个写数据请求。

5

16、根据权利要求 15 所述的存储系统，其特征在于，

所述数据节点，还用于为保存的写数据请求集合分配标识，将所述写数据请求的标识发送给所述校验计算节点；创建元数据，所述元数据包括所述写数据请求集合的标识与每个写数据请求中的数据待写入的逻辑地址之间
10 的对应关系，以及每个写数据请求中的数据待写入的逻辑地址与内部偏移量之间的对应关系。

17、根据权利要求 16 所述的存储系统，其特征在于，所述存储系统还包括元数据校验计算节点以及元数据校验节点，

15 所述数据节点，还用于在确定累积的元数据达到所述预设数据量时，将元数据集合以及数据节点的标识发送给所述元数据校验计算节点，所述元数据集合包括所述累积的达到所述预设数据量的元数据；

所述元数据校验计算节点，用于接收每个数据节点发送的元数据集合以及所述数据节点的标识，保存每个元数据集合与数据节点的标识之间的对应
20 关系，根据所述对应关系从接收的多个元数据集合中选择所述设定数量个元数据集合，所述设定数据量个元数据集合对应不同的数据节点的标识；计算所述设定数量个元数据集合的校验数据；将所述设定数量个元数据集合的校验数据发送给所述元数据校验节点，所述元数据校验节点不同于存储所述设定数量个元数据集合中的每个元数据集合的数据节点。

25

18、根据权利要求 16 所述的存储系统，其特征在于，所述存储系统还包括垃圾回收节点；所述校验计算节点，还用于为分条分配标识，所述分条

包括所述设定数量个写数据请求集合以及所述设定数量个写数据请求集合的校验数据；将所述分条标识与所述分条包括的每个写数据请求集合的标识之间的对应关系，发送给所述垃圾回收节点；

5 每个数据节点，还用于将保存的写数据请求集合的位图发送给所述垃圾回收节点，所述位图用于指示所述写数据请求集合的无效数据的数据量；

所述垃圾回收节点，用于根据所述分条标识与所述分条包括的每个写数据请求集合的标识之间的对应关系，以及每个写数据请求集合的位图确定所述分条包含的无效数据的数据量大于其他任意一个分条包含的无效数据的数据量；在确定所述分条包含的无效数据的数据量大于其他任意一个分条包含的无效数据的数据量时，向所述分条包含的每个写数据请求集合所在的数据节点分别发送垃圾回收通知消息，每个垃圾回收通知消息包括写数据请求集合的标识。

10

19、根据权利要求 18 所述的存储系统，其特征在于，

15 所述数据节点，还用于在接收所述垃圾回收通知消息之后，根据写数据请求集合的标识以及保存的所述写数据请求集合的位图，对所述写数据请求集合执行系统垃圾回收操作。

20、根据权利要求 14-19 任一所述的存储系统，其特征在于，所述设定数量是根据预先设定的校验模式确定的。

20

21、一种数据校验方法，其特征在于，所述方法应用于存储系统中，所述存储系统包括校验节点、校验计算节点和多个数据节点，每个数据节点具有唯一的标识，

25 每个数据节点将写数据请求集合发送给所述校验计算节点，所述写数据请求集合包括多个写数据请求，每个写数据请求包括数据以及所述数据待写入的数据节点的标识，所述写数据请求集合的大小等于预设数据量；

所述校验计算节点接收多个写数据请求集合；从所述多个写数据请求集合中选择设定数量个写数据请求集合，所述设定数量个写数据请求集合是所述多个写数据请求集合的子集，并且所述设定数量个写数据请求集合来自不同的数据节点；计算所述设定数量个写数据请求集合的校验数据；将所述校验数据发送给所述校验节点，所述校验节点不同于所述设定数量个写数据请求集合中的每个写数据请求集合所在的数据节点。

22、根据权利要求 21 所述的存储系统，其特征在于，每个写数据请求还包括所述数据待写入的逻辑地址，所述数据待写入的数据节点所述存储系统中的主机根据所述数据或者所述数据待写入的逻辑地址选择的；所述方法还包括：

每个数据节点接收所述主机发送的多个写数据请求。

23、根据权利要求 22 所述的方法，其特征在于，还包括：

所述数据节点为保存的写数据请求集合分配标识，将所述写数据请求集合的标识发送给所述校验计算节点；创建元数据，所述元数据包括所述写数据请求集合的标识与每个写数据请求中的数据待写入的逻辑地址之间的对应关系，以及每个写数据请求中的数据待写入的逻辑地址与内部偏移量之间的对应关系。

24、根据权利要求 23 所述的方法，其特征在于，所述存储系统还包括元数据校验计算节点以及元数据校验节点，所述方法还包括：

所述数据节点在确定累积的元数据达到所述预设数据量时，将元数据集合以及数据节点的标识发送给所述元数据校验计算节点，所述元数据集合包括所述累积的达到所述预设数据量的元数据；

所述元数据校验计算节点接收每个数据节点发送的元数据集合以及所述数据节点的标识，保存每个元数据集合与数据节点的标识之间的对应关

系, 根据所述对应关系从接收的多个元数据集合中选择所述设定数量个元数据集合, 所述设定数量个元数据集合对应不同的数据节点的标识; 计算所述设定数量个元数据集合的校验数据; 将所述设定数量个元数据集合的校验数据发送给所述元数据校验节点, 所述元数据校验节点不同于存储所述设定数量个元数据集合中每个元数据集合的数据节点。

25、根据权利要求 23 所述的方法, 其特征在于, 所述存储系统还包括垃圾回收节点, 所述方法还包括:

所述校验计算节点为分条分配标识, 所述分条包括所述设定数量个写数据请求集合以及所述设定数量个写数据请求集合的校验数据; 将所述分条标识与所述分条包括的每个写数据请求集合的标识之间的对应关系发送给垃圾回收节点;

每个数据节点将保存的写数据请求集合的位图发送给所述垃圾回收节点, 所述位图用于指示所述写数据请求集合的无效数据的数据量;

所述垃圾回收节点根据所述分条标识与所述分条包括的每个写数据请求集合的标识之间的对应关系, 以及每个写数据请求集合的位图确定所述分条包含的无效数据的数据量大于其他任意一个分条包含的无效数据的数据量; 在确定所述分条包含的无效数据的数据量大于其他任意一个分条包含的无效数据的数据量时, 向所述分条包含的每个写数据请求集合所在的数据节点分别发送垃圾回收通知消息, 每个垃圾回收通知消息包括写数据请求集合的标识。

26、根据权利要求 25 所述的方法, 其特征在于, 还包括:

所述数据节点在接收所述垃圾回收通知消息之后, 根据写数据请求集合的标识以及保存的所述写数据请求集合的位图, 对所述写数据请求集合执行系统垃圾回收操作。

27、根据权利要求 21-26 任一所述的方法, 其特征在于, 所述设定数量是根据预先设定的校验模式确定的。