

HDD 磁盘上文件元数据备份方法及元数据备份服务器

5 本申请要求于 2023 年 05 月 08 日提交中国专利局、申请号为 202310530686.4、申请名称为“HDD 磁盘上文件元数据备份方法及元数据备份服务器”的中国专利申请的优先权，其全部内容通过引用结合在本申请中。

技术领域

本申请涉及到云计算领域，具体而言，涉及一种 HDD 磁盘上文件元数据备份方法及元数据备份服务器。

10 背景技术

本部分旨在为权利要求书中陈述的本发明实施例提供背景或上下文。此处的描述不因为包括在本部分中就承认是现有技术。

对于存储系统来说，不丢不错是永恒的追求，但是在生产系统中，仍然面临各种原因的数据删除后找回需求，比如存储系统软件问题造成的文件误删、运维操作导致的文件误删或者文件系统损坏、用户误操作导致文件被删等等。为了守住数据不丢的底线，满足用户业务的需求，存储系统需要恢复这些被误删除的文件。对于 SSD（Solid State Disk 或 Solid State Drive，固态硬盘）闪存盘来说，全盘扫描不失为最后的保障，但是对于空间越来越大，但 IO（Input/Output，输入/输出）能力变化不大的 HDD（Hard Disk Drive，机械硬盘）磁盘来说，全盘扫描的时间需要按周计算，这严重影响了存储服务恢复的时效性，可能会对信息造成毁灭性的打击。为了加快数据恢复的速度，备份 HDD 上文件的元数据尤为重要。但现有的备份方案需要消耗较多的磁盘 IO 能力，这对于性能捉襟见肘的 HDD 磁盘来说，经常是不可接受的。

发明内容

25 本申请实施例提供了一种 HDD 磁盘上文件元数据备份方法及元数据备份服务器，可以在规避额外的磁盘 IO 操作的情况下，完成 HDD 磁盘上文件元数据的备份。

根据本申请的一个方面，还提供了一种 HDD 磁盘上文件元数据备份方法，包括：获取数据服务在所述 HDD 磁盘上写入数据的目录信息；监测所述目录信息的写入文件关闭事件，获取根据所述写入文件关闭事件返回的关闭文件信息；根据所述关闭文件信息确定关闭文件，获取所述关闭文件的元数据，以及备份所述元数据至元数据备份仓库。

30 根据本申请的另一个方面，还提供了一种元数据备份服务器，包括：获取模块、事件模块和备份模块：所述获取模块，用于获取数据服务在所述 HDD 磁盘上写入数据的目录信息；所述事件模块，用于监测所述目录信息的写入文件关闭事件，获取根据所述写入文

件关闭事件返回的关闭文件信息；所述备份模块，用于根据所述关闭文件信息确定关闭文件，获取所述关闭文件的元数据，以及备份所述元数据至元数据备份仓库。

根据本申请的另一个方面，还提供了一种存储系统，包括：上述的元数据备份服务器。

5 根据本申请的另一个方面，还提供了一种电子设备，包括：处理器；以及存储程序的存储器，其中，所述程序包括指令，所述指令在由所述处理器执行时使所述处理器执行根据上述的方法。

根据本申请的另一个方面，还提供了一种存储有计算机指令的非瞬时计算机可读存储介质，其中，所述计算机指令用于使所述计算机执行根据上述的方法步骤。

10 根据本申请的另一个方面，还提供了一种计算机程序产品，所述计算机程序产品包括计算机程序，所述计算机程序被处理器执行时实现上述的方法步骤。

本发明实施例的有益效果：

在本发明实施例中，获取数据服务在所述 HDD 磁盘上写入数据的目录信息；监测所述目录信息的写入文件关闭事件，获取根据所述写入文件关闭事件返回的关闭文件信息；根据所述关闭文件信息确定关闭文件，获取所述关闭文件的元数据，以及备份所述元数据
15 至元数据备份仓库。本发明在操作系统中监测数据服务在 HDD 磁盘上写入数据的目录信息，在监测到发生写入文件关闭事件时，获取根据所述写入文件关闭事件返回的关闭文件信息，根据所述关闭文件信息确定关闭文件，从而尽量在元数据还在内核内存时，获取关闭文件的元数据，在规避了额外的磁盘 IO 操作的情况下，实现备份所述元数据至元数据备份仓库。

20 本发明的一个或多个实施例的细节在以下附图和描述中提出，以使本发明的其他特征、目的和优点更加简明易懂。

附图说明

为了更清楚地说明本发明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发明
25 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实施例。

图 1 是本发明实施例的 HDD 磁盘上文件元数据备份方法的流程图；

图 2 是本发明实施例的常规备份和修正备份流程示意图；

图 3 是能够用于实现本发明的实施例的示例性电子设备的结构框图。

30 具体实施方式

下面将参照附图更详细地描述本发明实施例。虽然附图中显示了本发明的某些实施例，然而应当理解的是，本发明可以通过各种形式来实现，而且不应该被解释为限于这里阐述的实施例，相反提供这些实施例是为了更加透彻和完整地理解本发明。应当理解的是，本

发明的附图及实施例仅用于示例性作用，并非用于限制本发明的保护范围。

现有的 HDD 磁盘上文件元数据备份方案一，通过对文件全量/增量扫描实现元数据的备份。该方案定期的对盘上存储的文件进行扫描，然后逐个文件的调用 stat 获取文件的元数据信息，通过 fiemap 接口获取文件的 extent 信息，然后把每个文件的元数据/extent 信息持久化，从而完成文件元数据的全量备份或者自从上次备份以后修改文件元数据的增量备份。但是这种方法文件遍历顺序和磁盘存储方式解耦，所以需要大量的随机读操作，从而需要消耗过多的磁盘性能，对正常服务会造成影响。

现有的 HDD 磁盘上文件元数据备份方案二，是文件系统特异的元数据备份方案。实现该方案二的第一种方式是 inode table（一个文件系统中的数据结构，用于存储文件和目录的元数据信息，包括文件的大小、创建时间、修改时间、访问权限等）备份方式：对于常规的文件系统设计，元数据通常存储在相对连续的区域，为了降低元数据备份的磁盘 IO 数量，可以通过解析文件系统元数据区内容的方式完成元数据备份，比如 ext3 每个 inode 的存储空间占用 128 字节，ext4 每个 inode 的空间占用 256 字节，一次 64K 的 IO 操作可以获取 512 个 ext3 分区文件的元数据或 256 个 ext4 分区文件的元数据，从而相比方案一量级的节省磁盘 IO 的数量。

因为 HDD 磁盘的 IOPS（Input/Output Operations Per Second，每秒输入/输出操作数）上限比较低，而且存储数据普遍比较离散，所以方案一和方案二的第一种方式的随机读写比较频繁，元数据备份方式对 HDD 非常不友好。实现该方案二的第二种方式是 journal 区备份方式，通过解析文件系统 journal 区完成备份。因为 journal 区含有被修改元数据的备份，通过解析文件系统特异性的 journal 格式完成元数据备份。但 journal 区经常会保留对同一个文件的多次修改，另外存储效率低，浪费空间多，还经常面对 journal 预留空间可能较小的问题，从而对备份服务读 journal 时间窗口要求很高。另外不同文件系统的 journal 格式可能不一样，增加了方案适配的复杂度。

在常规的通过扫描文件或者备份文件系统元数据区、Journal 区的方案中，都不可避免的需要读取 HDD 盘上的数据，从而消耗磁盘 IO 能力，进而影响服务的性能。为了降低磁盘 IO 的操作，也可以通过增强内核完成元数据备份。比如通过内核模块或者 BPF（Berkeley Packet Filter，一种高效的网络数据包处理技术）等方案，增强内核功能，在内核中发现文件的元数据发生修改以后，持久化被修改文件的元数据，从而完成备份。但这种方案需要内核态的修改，所以对开发的要求非常高，稍有不慎引入 BUG 会有系统崩溃的可能，另外随着内核版本的变更迭代，也会引入不可控的适配风险。

基于以上背景，本申请现提出一种 HDD 磁盘上文件元数据备份方法及元数据备份服务器，该方法在写入文件的元数据还在内存的情况下，获取文件的元数据信息、extent 信息等，从而完成数据的备份，最终达到文件系统元数据备份的磁盘开销接近于无。

首先，对涉及的术语进行说明。

文件系统：操作系统用于存储设备或分区上的文件的方法和数据结构，即在存储设备

上组织文件的方法。

ext4 文件系统：第四代扩展文件系统（Fourth extended filesystem，ext4），Linux 系统下的日志文件系统，是 ext3 文件系统的后继版本。

元数据（Metadata）：又称中介数据、中继数据，为描述数据的数据，主要是描述数据属性的信息，用来支持如指示存储位置、历史数据、资源查找、文件记录等功能。

extent：是文件系统中的一种数据结构，用于管理文件在磁盘上的存储和检索。

fiemap（一种 Linux 文件系统的系统调用）：现代很多文件系统都采用了 extent 替代 block 来管理磁盘。extent 就是一些连续的 block，一个 extent 由起始的 block 加上长度进行定义。fiemap 是一种用户态获取文件存储 extent 信息的方法。

10 备份：容灾的基础，是指为防止系统出现操作失误或系统故障导致数据丢失，而将全部或部分数据集合从应用主机的硬盘或阵列复制到其它的存储介质的过程。

stat：是一个 Unix/Linux 系统调用，用于获取文件的元数据信息，包括文件的类型、大小、权限、创建时间、修改时间、访问时间等信息。通过调用 stat 函数，可以获取文件的元数据信息，并将其存储在一个结构体中。

15 inode：是文件系统中的一种数据结构，用于管理文件和目录的元数据信息。每个文件和目录都有一个对应的 inode，inode 包含了文件或目录的属性、权限、大小、创建时间、修改时间、访问时间等信息，但不包含文件的实际数据。

journal：是一种文件系统的日志记录机制，用于记录文件系统的操作和状态变化。journal 可以记录文件系统的元数据信息，包括文件的创建、修改、删除、移动等操作，以及文件系统的状态变化，如文件系统的挂载、卸载、恢复等操作。

20 inotify：是 Linux 内核提供了一种文件系统监控机制，它可以监控文件系统内的文件和目录，当文件或目录发生变化时，可以及时通知应用程序。

在本实施例中提供了一种 HDD 磁盘上文件元数据备份方法，图 1 是根据本申请实施例的 HDD 磁盘上文件元数据备份方法的流程图，下面对图 1 中所涉及到的方法步骤进行说明。

25 步骤 S102，获取数据服务在所述 HDD 磁盘上写入数据的目录信息。

在该步骤中，数据服务负责写入数据操作。例如，将需要存储的各种来源的数据写入 HDD 磁盘上的某个目录下的某个文件内。目录信息为这些被写入文件所在的目录。不同的文件信息可以属于同一个目录或者不同的目录。

30 需要说明的是，HDD 磁盘是计算机存储的主要形式之一，它们通常用于桌面计算机、笔记本电脑、服务器和其他设备中。

在该步骤中，HDD 磁盘上文件元数据备份启动后，获取数据服务写入数据的目录信息，以便对目录信息下的写入文件的状态进行监测。

35 步骤 S104，监测所述目录信息的写入文件关闭事件，获取根据所述写入文件关闭事件返回的关闭文件信息。

在该步骤中，写入文件关闭事件，用于表征写入文件完成本次写入或修改，写入文件被关闭。监测目录信息对应的各个写入文件的状态，当监测到写入文件关闭事件时，操作系统会根据写入文件关闭事件，返回被关闭的文件相关的描述信息和识别信息，即关闭文件信息。获取该根据所述写入文件关闭事件返回的关闭文件信息。

5 需要说明的是，写入文件关闭事件指的是在文件写入完成后，关闭文件的操作。关闭文件的操作可以包括以下几个方面：刷新缓冲区、释放文件资源和确保数据完整性。关闭文件的操作，可以确保数据的持久化存储和文件资源的释放。

另外需要说明的是，上述被关闭的文件相关的描述信息中可以包括该被关闭的文件的修改内容信息。上述被关闭的文件相关的识别信息，可以用于识别以及查询到该被关闭的
10 文件。

在该步骤中，在监测到目录信息的写入文件关闭事件时，此时，文件虽然被关闭，但是元数据还在操作系统的内存中，因此，获取关闭文件信息，以便后续基于文件信息获取元数据，在关闭时间点马上进行元数据查询不会产生额外的磁盘 IO。

步骤 S106，根据所述关闭文件信息确定关闭文件，获取所述关闭文件的元数据，以及
15 备份所述元数据至元数据备份仓库。

在该步骤中，在得到关闭文件信息后，即可确定出被关闭的写入文件，即关闭文件。在该关闭文件的元数据还在内存的情况下，可以实时地通过标准的操作系统接口获取所述关闭文件的元数据，从而完成数据的备份。

需要说明的是，文件的元数据是指描述文件属性、特征、结构、格式、来源、质量等
20 信息的数据。常见的文件元数据包括：文件名：文件的名称，用于标识文件。文件类型：文件的类型，如文本文件、图像文件、音频文件、视频文件等。文件大小：文件的大小，以字节、千字节、兆字节等单位表示。创建时间和修改时间：文件的创建时间和最后修改时间。文件路径：文件在文件系统的路径，用于标识文件的位置。文件权限：文件的访问权限，包括读、写、执行等权限。文件格式：文件的格式，如文本文件的编码格式、图像文件的压缩格式等。文件版本：文件的版本号，用于标识文件的不同版本。文件摘要：
25 文件的摘要信息，如文件的哈希值、数字签名等。

另外需要说明的是，元数据备份仓库可以是本地磁盘、网络存储设备或云存储服务，具体采用的元数据备份仓库可以根据实际需求进行选择，本发明实施例对此不作具体限定。将系统元数据备份到元数据备份仓库，以便在系统故障或数据丢失时进行恢复。

30 在本发明实施例中，获取数据服务在所述 HDD 磁盘上写入数据的目录信息；监测所述目录信息的写入文件关闭事件，获取根据所述写入文件关闭事件返回的关闭文件信息；根据所述关闭文件信息确定关闭文件，获取所述关闭文件的元数据，以及备份所述元数据至元数据备份仓库。本发明在操作系统中监测数据服务在 HDD 磁盘上写入数据的目录信息，在监测到发生写入文件关闭事件时，获取根据所述写入文件关闭事件返回的关闭文件
35 信息，根据所述关闭文件信息确定关闭文件，从而尽量在元数据还在内核内存时，获取关

闭文件的元数据，在规避了额外的磁盘 IO 操作的情况下，实现备份所述元数据至元数据备份仓库。

在一种可能的实施方式中，监测所述目录信息的写入文件关闭事件，获取根据所述写入文件关闭事件返回的关闭文件信息，可以按照如下步骤执行：利用 inotify 组件，监测所述目录信息的写入文件关闭事件，获取所述 inotify 组件根据所述写入文件关闭事件返回的关闭文件信息；其中，所述关闭文件信息包括监控描述符和文件名。

在本发明实施例中，inotify 可以帮助应用程序实时监控文件系统的变化，提高应用程序的效率和可靠性。利用 inotify 组件，监测所述目录信息的写入文件关闭事件，当监测到写入文件关闭事件时，可以及时地获取所述 inotify 组件根据所述写入文件关闭事件返回的关闭文件信息。

关闭文件信息包括监控描述符和文件名。其中，监控描述符（Watch Descriptor，WD）是 inotify 机制中的一个概念，它表示一个被监控的文件或目录。在使用 inotify 机制时，应用程序可以使用 inotify_init() 函数创建一个 inotify 实例，并使用 inotify_add_watch() 函数添加一个监控目录，返回代表该目录的监控描述符。监控描述符和文件名可以用于识别关闭文件的完整路径并唯一定位修改的文件。

需要说明的是，inotify 可以用于：文件系统监控：inotify 可以监控文件系统中的文件和目录，包括文件的创建、修改、删除、移动等操作。当文件或目录发生变化时，inotify 会及时通知应用程序。实时通知：inotify 可以实时通知应用程序文件系统的变化，避免了轮询文件系统的开销和延迟。精确监控：inotify 可以精确监控文件系统中的文件和目录，可以指定监控的文件类型、目录层级、文件名等。应用场景广泛：inotify 可以应用于很多场景，如文件同步、日志监控、安全审计等。

在一种可能的实施方式中，根据所述关闭文件信息确定关闭文件，可以按照如下步骤执行：根据所述监控描述符和所述文件名生成关闭文件路径；根据所述关闭文件路径确定关闭文件。

在本发明实施例中，利用监控描述符和文件名进行拼接，可以得到关闭文件路径，关闭文件路径可以用于查找到关闭文件。

在该步骤中，可以基于关闭文件路径获得关闭文件，为基于关闭文件获取元数据提供支持。

在一种可能的实施方式中，获取所述关闭文件的元数据，可以按照如下步骤执行：利用 fiemap 接口和/或 stat 系统调用，获取所述关闭文件的元数据。

在本发明实施例中，fiemap 接口和 stat 都可以用于获取文件系统的元数据信息，但是它们的使用场景略有不同。

fiemap 接口主要用于获取文件在磁盘上的物理位置信息，可以用于实现文件碎片整理、文件压缩等操作。fiemap 可以获取文件的所有物理块的位置信息，包括稀疏文件中的空洞块。fiemap 的使用需要文件系统支持。

而 stat 系统调用则主要用于获取文件的基本属性信息，如文件大小、创建时间、修改时间、访问时间等。stat 可以获取文件的 inode 号，可以用于判断两个文件是否为同一个文件。stat 的使用不需要 root 权限，可以适用于普通用户。

5 综上所述可知，如果需要获取文件的物理位置信息，可以使用 fiemap 接口；如果需要获取文件的基本属性信息，可以使用 stat 调用，也可根据实际需求，同时采用该两种方式获取关闭文件的元数据，具体选择何种方式获取元数据，可以根据实际需求进行确定，本发明实施例对此不作具体限定。

考虑到常规备份路径依赖于备份服务能够感知所有文件的修改，并且正常持久化这些修改文件的元数据，但是实际操作中，inotify 监视的事件有可能丢失，并且元数据备份服务有可能在某个时刻突然被杀或者异常退出，造成期间修改文件的元数据没有正常备份，10 这种情况下，就需要单独的修正备份通道进行查漏补缺，因此，在一种可能的实施方式中，备份所述元数据后，还可以执行如下步骤：获取所述数据服务管理的第一文件的第一元数据；其中，所述第一元数据包括文件名数据、创建时刻数据、Inode 数据和文件长度数据中的一种或几种；加载所述元数据备份仓库中的第二元数据；对比所述第一元数据和所述15 第二元数据，根据对比结果确定增量文件列表信息，根据所述增量文件列表信息，获取增量元数据，以及备份所述增量元数据。

在本发明实施例中，将数据服务正在管理的所有文件作为第一文件，将数据服务中的第一文件的元数据作为第一元数据。第一元数据中包括多种类型的元数据，可以获取其中文件名数据、创建时刻数据、Inode 数据和文件长度数据中的一种或几种。将元数据备份20 仓库中所有的元数据作为第二元数据。

需要说明的是，上述获取第一文件的第一元数据以及加载元数据备份仓库中的第二元数据的两个步骤是两个独立的步骤，二者的执行顺序不分先后，二者可以同时执行，或者择一优先执行，可以根据实际需求确定执行该二者的先后顺序，本发明实施例对此不作具体限定。

25 对比所述第一元数据和所述第二元数据，将得到的差异信息作为对比结果，根据对比结果确定增量文件列表信息。其中，增量文件列表信息包括多个文件名信息。在得到增量文件列表信息后，按照类似于步骤 S106 的方法，获取多个需要重新进行备份的文件的元数据，即增量元数据，再将增量元数据补充备份至元数据备份仓库，使元数据备份仓库中的备份元数据与数据服务管理的文件的元数据相对应，避免元数据备份仓库中的备份元数30 据存在错漏的情况，降低正常备份流程情况下，各种异常问题情况对元数据备份结果的影响。

在该步骤中，通过补充备份增量元数据，可以对常规备份进行查漏补缺，保证元数据的完整性和准确性。

35 在一种可能的实施方式中，根据对比结果确定增量文件列表信息，可以按照如下步骤执行：若所述第一元数据包括第一目标元数据，且所述第二元数据不包括所述第一目标元

数据，则将所述第一目标元数据对应的文件名数据添加至所述增量文件列表信息；若所述第一元数据中的第二目标元数据，与所述第二目标元数据在所述第二元数据中的文件名数据、创建时刻数据、Inode 数据和文件长度数据中任一项或几项不同，则将所述第一元数据中所述第二目标元数据的文件名添加至所述增量文件列表信息。

- 5 在该步骤中，若某文件对应的第一目标元数据仅存在于第一元数据，不存在于第二元数据中，则说明该文件可能在备份的时候被漏掉或者备份失败，因此，需要将第一目标元数据对应的上述文件的文件名数据添加至所述增量文件列表信息，以便将该文件的元数据补充备份至元数据备份仓库。

- 10 需要说明的是，第一目标元数据是第一元数据中的一个子集。第一目标元数据是（或者不是）第二元数据中某个子集的一部分数据。

另外，若备份过程正常进行，则第一元数据中的第二目标元数据包含的文件名数据、创建时刻数据、Inode 数据和文件长度数据与第二目标元数据在第二元数据中包含的这些数据相同，若这些数据中任意一项或几项不相同，则将第一元数据中第二目标元数据的文件名添加至所述增量文件列表信息。

- 15 需要说明的是，第二目标元数据是某个文件的元数据，若正常备份，则第二目标元数据在第一元数据中和在第二元数据中是一样的，若不一样，则证明备份过程可能存在错误，则以第一元数据中的内容为准，重新对该文件的元数据进行备份。

- 20 在该步骤中，文件名数据、创建时刻数据、Inode 数据和文件长度数据等元数据信息，需要的存储空间比较少，在存储介质中连续存储，同时可以存储在高性能存储设备中，从而读取开销也比较低。另外，第一文件的第一元数据也可以在内存态通过对数据服务远程过程调用获取，从而对 HDD 盘的 IO 性能无损耗。

在一种可能的实施方式中，获取所述数据服务管理的第一文件的第一元数据，可以按照如下步骤执行：通过与所述数据服务远程过程调用和/或加载所述数据服务的 checkpoint 文件获取所述数据服务管理的第一文件的第一元数据。

- 25 在本发明实施例中，远程过程调用（Remote Procedure Call，RPC）是一种计算机通信协议，用于在不同的计算机之间或者同计算机的不同进程间进行远程调用。在 RPC 中，调用方（客户端）向被调用方（服务器）发送请求消息，请求消息中包含要调用的远程过程的名称和参数。服务器接收到请求消息后，执行相应的远程过程，并将结果返回给客户端。客户端接收到结果后，继续执行本地程序。RPC 可以使用不同的传输协议和编码方式。

- 30 checkpoint 文件是一种用于保存进程状态的文件，通常用于实现进程的恢复和迁移。当一个进程需要被暂停或迁移时，它的状态信息会被写入 checkpoint 文件中，包括进程的内存映像、寄存器状态、文件描述符等信息。checkpoint 文件可以在另一个系统上被读取和恢复，使得进程可以在不同的系统上继续执行。

- 35 在该步骤中，通过远程过程调用或 checkpoint，或同时采用远程过程调用和 checkpoint 文件，确定所述数据服务正在管理的第一文件的第一元数据，从而为获得增量元数据提供

支持。

参见图 2 所示的常规备份和修正备份流程示意图，下面以一个具体实施例对该方法可能的实施方式进行说明。为了实现接近无额外磁盘 IO 的元数据备份服务，可以通过常规备份和查漏补缺的修正备份两个通道完成元数据的备份。其中，

5 1.常规备份：

a.备份服务启动后，获取数据服务写入数据的目录，然后通过 inotify_add_watch 注册对这些目录下写入文件关闭事件的监控；

10 b.在操作系统发现监控目录下文件写 fd（文件描述符，是操作系统中用于标识和访问文件或其他输入/输出资源的整数值）关闭之后，会通过 inotify_event 的方式返回元数据备份服务被关闭文件的监控描述符(wd)，以及被关闭的文件名。元数据备份服务器可以基于这两个信息拼接出完成的关闭文件路径；

15 c.在文件系统中，可以通过 stat 获取文件的基础元数据，比如文件的长度、创建/修改时间等等。另外操作系统也提供了 fiemap 的接口，可以在用户态非常高效的获取文件系统上某个文件的存储位置。常规情况下，步骤 1b 中的文件虽然被关闭，但是元数据还在操作系统的内存中，所以在关闭时间点马上进行元数据查询不会产生额外的磁盘 IO。

2.修正备份：

20 步骤 1 的常规备份路径依赖于备份服务能够感知所有文件的修改，并且正常持久化这些修改文件的元数据，但是实际操作中，inotify 监视的事件有可能丢失，并且元数据备份服务有可能在某个时刻突然被杀或者异常退出，造成期间修改或者新建文件的元数据没有正常备份，这种情况下，就需要单独的修正备份通道进行查漏补缺。基本流程如下：

a.通过 RPC 或者数据服务的 checkpoint 文件获取当前数据服务中有效的文件列表和必要的元数据，比如文件长度、创建时间、Inode 号等。checkpoint 中存储信息需要的存储空间比较少，在存储介质中连续存储，从而读取开销也比较低，另外也可以在内存态通过 RPC 获取，从而对 HDD 盘的 IO 性能无损耗

25 b.读取元数据备份仓库中已知的备份元数据，构建已知文件元数据备份表

c.比较步骤 2a 和步骤 2b 中的条目，如果同时满足如下条件，则认为备份的元数据仍然有效，否则通过可能消耗磁盘 IO 的类步骤 1c 的方式，完成元数据备份的查漏补缺：i. 文件名相同；ii.创建时间相同；iii.Inode 号相同；iv.文件长度相同。

30 上述元数据备份过程，通过 stat/fiemap 获取到足够的信息后，可以把包括但不限于文件名（文件路径）、文件长度、文件创建时间、Inode 号、extents 列表等信息持久化，从而完成元数据备份。

通过如上的方案，可以在常规路径基本无额外 HDD 磁盘 IO 的情况下，完成元数据的备份，从而降低了对生产服务的影响，提高了备份服务的可用性。另外，这种备份方式，不需要在存储服务中做任何修改，所以适用性更广。

35 本申请提供了一种 HDD 磁盘上文件元数据备份方法，该方法提供了通用的文件系统

元数据备份方案，可以适用于多种云上和云下的存储系统。该方法通过操作系统接口，监视数据目录下文件写句柄关闭的事件，以及之后进行查漏补缺的修正方案，能够尽可能的确保相应元数据还在内核内存的时间点查找，从而规避了额外的磁盘 IO 操作。另外所有操作都是使用的操作系统支持的常规接口，不需要内核做出适配，从而极大地降低各种风险。

5 本申请还提供了一种元数据备份服务器，包括：获取模块、事件模块和备份模块：所述获取模块，用于获取数据服务在所述 HDD 磁盘上写入数据的目录信息；所述事件模块，用于监测所述目录信息的写入文件关闭事件，获取根据所述写入文件关闭事件返回的关闭文件信息；所述备份模块，用于根据所述关闭文件信息确定关闭文件，获取所述关闭文件的元数据，以及备份所述元数据至元数据备份仓库。

10 在本申请实施例中，所述获取模块，用于获取数据服务在所述 HDD 磁盘上写入数据的目录信息；所述事件模块，用于监测所述目录信息的写入文件关闭事件，获取根据所述写入文件关闭事件返回的关闭文件信息；所述备份模块，用于根据所述关闭文件信息确定关闭文件，获取所述关闭文件的元数据，以及备份所述元数据至元数据备份仓库。本发明
15 实施例在操作系统中监测数据服务在 HDD 磁盘上写入数据的目录信息，在监测到发生写入文件关闭事件时，获取根据所述写入文件关闭事件返回的关闭文件信息，根据所述关闭文件信息确定关闭文件，从而尽量在元数据还在内核内存时，获取关闭文件的元数据，在规避了额外的磁盘 IO 操作的情况下，实现备份所述元数据至元数据备份仓库。

20 在一种可能的实施方式中，所述事件模块，具体用于：利用 inotify 组件，监测所述目录信息的写入文件关闭事件，获取所述 inotify 组件根据所述写入文件关闭事件返回的关闭文件信息；其中，所述关闭文件信息包括监控描述符和文件名。

在一种可能的实施方式中，所述备份模块，具体用于：根据所述监控描述符和所述文件名生成关闭文件路径；根据所述关闭文件路径确定关闭文件。

25 在一种可能的实施方式中，所述备份模块，具体用于：利用 fiemap 接口和/或 stat 系统调用，获取所述关闭文件的元数据。

30 在一种可能的实施方式中，还包括修正模块，用于：获取所述数据服务管理的第一文件的第一元数据；其中，所述第一元数据包括文件名数据、创建时刻数据、Inode 数据和文件长度数据中的一种或几种；加载所述元数据备份仓库中的第二元数据；对比所述第一元数据和所述第二元数据，根据对比结果确定增量文件列表信息；根据所述增量文件列表信息，获取增量元数据，以及备份所述增量元数据。

35 在一种可能的实施方式中，所述修正模块，具体用于：若所述第一元数据包括第一目标元数据，且所述第二元数据不包括所述第一目标元数据，则将所述第一目标元数据对应的文件名数据添加至所述增量文件列表信息；若所述第一元数据中的第二目标元数据，与所述第二目标元数据在所述第二元数据中的文件名数据、创建时刻数据、Inode 数据和文件长度数据中任一项或几项不同，则将所述第一元数据中所述第二目标元数据的文件名添

加至所述增量文件列表信息。

在一种可能的实施方式中，所述修正模块，具体用于：通过与所述数据服务远程过程调用和/或加载所述数据服务的 checkpoint 文件获取所述数据服务管理的第一文件的第一元数据。

5 本发明实施例还提供一种存储系统，包括：上述的元数据备份服务器。

本发明实施例还提供一种电子设备，包括：至少一个处理器；以及与至少一个处理器通信连接的存储器。所述存储器存储有能够被所述至少一个处理器执行的计算机程序，所述计算机程序在被所述至少一个处理器执行时用于使所述电子设备执行根据本发明实施例的方法。

10 本发明实施例还提供一种存储有计算机程序的非瞬时计算机可读存储介质，其中，所述计算机程序在被计算机的处理器执行时用于使所述计算机执行根据本发明实施例的方法。

本发明实施例还提供一种计算机程序产品，包括计算机程序，其中，所述计算机程序在被计算机的处理器执行时用于使所述计算机执行根据本发明实施例的方法。

15 参考图 3，现将描述可以作为本发明的服务器或用户的电子设备 300 的结构框图，其是可以应用于本发明的各方面的硬件设备的示例。电子设备旨在表示各种形式的数字电子的计算机设备，诸如，膝上型计算机、台式计算机、工作台、个人数字助理、服务器、刀片式服务器、大型计算机、和其它适合的计算机。电子设备还可以表示各种形式的移动装置，诸如，个人数字处理、蜂窝电话、智能电话、可穿戴设备和其它类似的计算装置。本文所示的部件、它们的连接和关系、以及它们的功能仅仅作为示例，并且不意在限制本文中描述的和/或者要求的本发明的实现。

20 如图 3 所示，电子设备 300 包括计算单元 301，其可以根据存储在只读存储器 (ROM) 302 中的计算机程序或者从存储单元 308 加载到随机访问存储器 (RAM) 303 中的计算机程序，来执行各种适当的动作和处理。在 RAM 303 中，还可存储设备 300 操作所需的各种程序和/或数据。计算单元 301、ROM 302 以及 RAM 303 通过总线 304 彼此相连。输入/输出 (I/O) 接口 305 也连接至总线 304。

25 电子设备 300 中的多个部件连接至 I/O 接口 305，包括：输入单元 306、输出单元 307、存储单元 308 以及通信单元 309。输入单元 306 可以是能向电子设备 300 输入信息的任何类型的设备，输入单元 306 可以接收输入的数字或字符信息，以及产生与电子设备的用户设置和/或功能控制有关的键信号输入。输出单元 307 可以是能呈现信息的任何类型的设备，并且可以包括但不限于显示器、扬声器、视频/音频输出终端、振动器和/或打印机。存储单元 308 可以包括但不限于磁盘、光盘。通信单元 309 允许电子设备 300 通过诸如因特网的计算机网络和/或各种电信网络与其他设备交换信息/数据，并且可以包括但不限于调制解调器、网卡、红外通信设备、无线通信收发机和/或芯片组，例如蓝牙设备、WiFi 设备、30 WiMax 设备、蜂窝通信设备和/或类似物。

计算单元 301 可以是各种具有处理和计算能力的通用和/或专用处理组件。计算单元 301 的一些示例包括但不限于处理单元 (CPU)、图形处理单元 (GPU)、各种专用的人工智能 (AI) 计算芯片、各种运行机器学习模型算法的计算单元、数字信号处理器 (DSP)、以及任何适当的处理器、控制器、微控制器等。计算单元 301 执行上文所描述的各个方法和处理。例如, 在一些实施例中, 前述的方法可被实现为计算机软件程序, 其被有形地包含于机器可读介质, 例如存储单元 308。在一些实施例中, 计算机程序的部分或者全部可以经由 ROM302 和/或通信单元 309 而被载入和/或安装到电子设备 300 上。在一些实施例中, 计算单元 301 可以通过其他任何适当的方式 (例如, 借助于固件) 而被配置为执行前述的方法。

10 用于实施本发明实施例的方法的程序代码可以采用一个或多个编程语言的任何组合来编写。这些程序代码可以提供给通用计算机、专用计算机或其他可编程数据处理装置的处理器或控制器, 使得程序代码当由处理器或控制器执行时使流程图和/或框图中所规定的功能/操作被实施。程序代码可以完全在机器上执行、部分地在机器上执行, 作为独立软件包部分地在机器上执行且部分地在远程机器上执行或完全在远程机器或服务器上执行。

15 在本发明实施例的上下文中, 机器可读介质可以是有形的介质, 其可以包含或存储以供指令执行系统、装置或设备使用或与指令执行系统、装置或设备结合地使用的程序。机器可读介质可以是机器可读信号介质或机器可读储存介质。机器可读介质可以包括但不限于电子的、磁性的、光学的、电磁的、红外的、或半导体系统、装置或设备, 或者上述内容的任何合适组合。机器可读存储介质的更具体示例会包括基于一个或多个线的电气连接、
20 便携式计算机盘、硬盘、随机存取存储器 (RAM)、只读存储器 (ROM)、可擦除可编程只读存储器 (EPROM 或快闪存储器)、光纤、便捷式紧凑盘只读存储器 (CD-ROM)、光学储存设备、磁储存设备、或上述内容的任何合适组合。

需要说明的是, 本发明实施例使用的术语“包括”及其变形是开放性包括, 即“包括但不限于”。术语“基于”是“至少部分地基于”。术语“一个实施例”表示“至少一个实施例”; 术语“另一实施例”表示“至少一个另外的实施例”; 术语“一些实施例”表示“至少一些实施例”。本发明实施例中提及的“一个”、“多个”的修饰是示意性而非限制性的, 本领域技术人员应当理解, 除非在上下文另有明确指出, 否则应该理解为“一个或多个”。

本发明实施例所涉及的用户信息 (包括但不限于用户设备信息、用户个人信息等) 和数据 (包括但不限于用于分析的数据、存储的数据、展示的数据等), 均为经用户授权或者经过各方充分授权的信息和数据, 并且相关数据的收集、使用和处理需要遵守相关国家和地区的相关法律法规和标准, 并提供有相应的操作入口, 供用户选择授权或者拒绝。

本发明实施例所提供的方法实施方式中记载的各个步骤可以按照不同的顺序执行, 和/或并行执行。此外, 方法实施方式可以包括附加的步骤和/或省略执行示出的步骤。本发明的保护范围在此方面不受限制。

35 “实施例”一词在本说明书中指的是结合实施例描述的具体特征、结构或特性可以包括

在本发明的至少一个实施例中。该短语出现在说明书中的各个位置并不一定意味着相同的实施例，也不意味着与其它实施例相互排斥而具有独立性或可供选择。本说明书中的各个实施例均采用相关的方式描述，各个实施例之间相同相似的部分互相参见。尤其，对于装置、设备、系统实施例而言，由于其基本相似于方法实施例，所以描述的比较简单，相关之处参见方法实施例的部分说明。

5

以上所述实施例仅表达了本发明的几种实施方式，其描述较为具体和详细，但并不能因此而理解为对专利保护范围的限制。应当指出的是，对于本领域的普通技术人员来说，在不脱离本发明构思的前提下，还可以做出若干变形和改进，这些都属于本发明的保护范围。因此，本发明的保护范围应以所附权利要求为准。

权 利 要 求 书

1. 一种 HDD 磁盘上文件元数据备份方法，包括：

获取数据服务在所述 HDD 磁盘上写入数据的目录信息；

5 监测所述目录信息的写入文件关闭事件，获取根据所述写入文件关闭事件返回的关闭文件信息；

根据所述关闭文件信息确定关闭文件，获取所述关闭文件的元数据，以及备份所述元数据至元数据备份仓库。

2. 根据权利要求 1 所述的方法，其中，监测所述目录信息的写入文件关闭事件，获取根据所述写入文件关闭事件返回的关闭文件信息，包括：

10 利用 inotify 组件，监测所述目录信息的写入文件关闭事件，获取所述 inotify 组件根据所述写入文件关闭事件返回的关闭文件信息；其中，所述关闭文件信息包括监控描述符和文件名。

3. 根据权利要求 2 所述的方法，其中，根据所述关闭文件信息确定关闭文件，包括：

根据所述监控描述符和所述文件名生成关闭文件路径；

15 根据所述关闭文件路径确定关闭文件。

4. 根据权利要求 1 所述的方法，其中，获取所述关闭文件的元数据，包括：

利用 fiemap 接口和/或 stat 系统调用，获取所述关闭文件的元数据。

5. 根据权利要求 1-4 任一项所述的方法，其中，备份所述元数据后，还包括：

20 获取所述数据服务管理的第一文件的第一元数据；其中，所述第一元数据包括文件名数据、创建时刻数据、Inode 数据和文件长度数据中的一种或几种；

加载所述元数据备份仓库中的第二元数据；

对比所述第一元数据和所述第二元数据，根据对比结果确定增量文件列表信息；

根据所述增量文件列表信息，获取增量元数据，以及备份所述增量元数据。

6. 根据权利要求 5 所述的方法，其中，根据对比结果确定增量文件列表信息，包括：

25 若所述第一元数据包括第一目标元数据，且所述第二元数据不包括所述第一目标元数据，则将所述第一目标元数据对应的文件名数据添加至所述增量文件列表信息；

若所述第一元数据中的第二目标元数据，与所述第二目标元数据在所述第二元数

据中的文件名数据、创建时刻数据、Inode 数据和文件长度数据中任一项或几项不同，则将所述第一元数据中所述第二目标元数据的文件名添加至所述增量文件列表信息。

7. 根据权利要求 5 所述的方法，其中，获取所述数据服务管理的第一文件的第一元数据，包括：

5 通过与所述数据服务远程过程调用和/或加载所述数据服务的 checkpoint 文件获取所述数据服务管理的第一文件的第一元数据。

8. 一种元数据备份服务器，包括：获取模块、事件模块和备份模块：

所述获取模块，用于获取数据服务在所述 HDD 磁盘上写入数据的目录信息；

10 所述事件模块，用于监测所述目录信息的写入文件关闭事件，获取根据所述写入文件关闭事件返回的关闭文件信息；

所述备份模块，用于根据所述关闭文件信息确定关闭文件，获取所述关闭文件的元数据，以及备份所述元数据至元数据备份仓库。

9. 一种存储系统，包括：权利要求 8 所述的元数据备份服务器。

10. 一种电子设备，包括：

15 处理器；以及

存储程序的存储器，

其中，所述程序包括指令，所述指令在由所述处理器执行时使所述处理器执行根据权利要求 1-7 任一项所述的方法步骤。

20 11. 一种存储有计算机指令的非瞬时计算机可读存储介质，其中，所述计算机指令用于使所述计算机执行根据权利要求 1-7 任一项所述的方法步骤。

12. 一种计算机程序产品，其中，所述计算机程序产品包括计算机程序，所述计算机程序被处理器执行时实现权利要求 1-7 任一项所述的方法步骤。