

存储卷挂载与访问方法、设备及存储介质

本申请要求于 2023 年 04 月 03 日提交中国专利局、申请号为 CN202310369529.X、发明名称为“存储卷挂载与访问方法、设备及存储介质”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。

5 存储卷挂载与访问方法、设备及存储介质

技术领域

本申请涉及云存储技术领域，尤其涉及一种存储卷挂载与访问方法、设备及存储介质。

背景技术

在云计算场景中，容器化应用部署形态已经成为了主流的应用部署方式，也是云原生
10 (Cloud Native) 计算变革的重要组成部分，大量应用逐步在进行容器化改造。为了实现容器化应用中数据的持久化存储，提出存储卷 (Volume) 的概念，这为容器化应用使用宿主机的存储系统作为其数据共享和持久化的存储设施提供了基础。那么，如何通过存储卷让容器化应用使用宿主机的存储系统作为其数据共享和持久化的存储设施，是云计算领域面临的一大技术问题。

15 发明内容

本申请的多个方面提供一种存储卷挂载与访问方法、设备及存储介质，用以以子树挂载方式将存储卷挂载到宿主机目录树，使得应用实例能够通过存储卷使用宿主机的存储系统作为数据共享和持久化的存储设施。

本申请实施例提供一种存储卷挂载方法，包括：确定被目标应用实例绑定且待挂载到
20 宿主机目录树上的目标存储卷，所述宿主机目录树是所述目标应用实例所在宿主机的目录树；若所述宿主机目录树上包含所述目标存储卷的根挂载点，针对所述目标应用实例在所述宿主机目录树中创建指向所述根挂载点的子挂载点；基于所述子挂载点以子树挂载方式将所述目标存储卷挂载到所述根挂载点上，以使所述目标应用实例通过所述根挂载点访问所述目标存储卷。

25 本申请实施例还提供一种存储卷访问方法，包括：响应目标应用实例对目标存储卷的访问操作，所述目标存储卷是被所述目标应用实例绑定且挂载到宿主机目录树上的存储卷；根据所述目标存储卷在所述宿主机目录树上的根挂载点和子挂载点，生成所述目标存储卷的访问路径；根据所述访问路径，对所述目标存储卷执行相应类型的访问操作。

30 本申请实施例还提供一种计算机设备，包括：存储器和处理器；所述存储器，用于存储计算机程序；所述处理器耦合至所述存储器，用于执行所述计算机程序以用于执行本申请实施例提供的存储卷挂载方法和/或存储卷访问方法中的步骤。

本申请实施例还提供一种存储有计算机程序的计算机可读存储介质，当所述计算机程

序被处理器执行时，致使所述处理器能够实现本申请实施例提供的存储卷挂载方法和/或存储卷访问方法中的步骤。

在本申请实施例中，针对被应用实例绑定的存储卷，以子树挂载方式将存储卷挂载到宿主机目录树上，即在宿主机目录树上包括存储卷的根挂载点和指向根挂载点的子挂载点，使得应用实例能够通过宿主机目录树上的根挂载点访问存储卷，进而达到通过存储卷使用宿主机的存储系统作为应用实例实现数据共享和持久化的存储设施的目的。

进一步，以子树挂载存储卷的方式，在存储卷被多个应用实例绑定的情况下，所有应用实例均通过根挂载点访问存储卷，能够共享相同的缓存空间，节约宿主机的内存资源。

进一步，子树挂载方式属于宿主机操作系统原生支持的挂载操作，在系统级的信息统计、挂载权限控制方面能够保持与独立挂载一样的逻辑和使用体验。

附图说明

此处所说明的附图用来提供对本申请的进一步理解，构成本申请的一部分，本申请的示意性实施例及其说明用于解释本申请，并不构成对本申请的不当限定。在附图中：

图1为本申请实施例提供的一种存储卷挂载方法的流程图；

图2为本申请实施例提供的另一种存储卷挂载方法的流程图；

图3为本申请实施例提供的一种存储卷访问方法的流程图；

图4为本申请实施例提供的另一种存储卷访问方法的流程图；

图5为本申请实施例提供的一种K8s集群中挂载PV的场景示意图；

图6为本申请实施例提供的一种存储卷挂载装置的结构示意图；

图7为本申请实施例提供的一种存储卷访问装置的结构示意图；

图8为本申请实施例提供的一种计算机设备的结构示意图。

具体实施方式

为使本申请的目的、技术方案和优点更加清楚，下面将结合本申请具体实施例及相应的附图对本申请技术方案进行清楚、完整地描述。显然，所描述的实施例仅是本申请一部分实施例，而不是全部的实施例。基于本申请中的实施例，本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例，都属于本申请保护的范围。

在本申请的实施例中，“至少一个”是指一个或者多个，“多个”是指两个或两个以上。“和/或”，描述关联对象的访问关系，表示可以存在三种关系，例如，A和/或B，可以表示：单独存在A，同时存在A和B，单独存在B这三种情况，其中A，B可以是单数或者复数。在本申请的文字描述中，字符“/”一般表示前后关联对象是一种“或”的关系。此外，在本申请实施例中，“第一”、“第二”、“第三”、等只是为了区分不同对象的内容而已，并无顺序、数量等其它特殊含义。

下面对本申请实施例中涉及的一些名词进行解释：

Kubernetes（简称 K8s）集群：是一种开源的用于自动部署、扩展和管理容器化应用的容器编排管理平台。

应用实例：是云计算环境中应用程序（简称应用）运行的最小单元。应用通常在一个或者多个应用实例上运行，应用可以拥有多个部署版本，不同应用实例上可以运行着不同版本的应用程序。应用实例的实例规格例如可以通过中央处理器（Central Processing Unit，CPU）核数和/或内存容量定义。例如，一种实例规格为 8 核 CPU 和 8GB(Gigabyte，十亿字节)的内存；另一种实例规格为 4 核 CPU 和 4GB(Gigabyte，十亿字节)的内存。

宿主机目录树：是指宿主机的操作系统级的目录树。本实施例并不限定宿主机使用的操作系统，例如包括但不限于：Linux 或 Unix。以 Linux 操作系统的目录树为例，在 Linux 下一切皆为文件，Linux 操作系统中的所有文件存储在文件系统中，它们被组织到一个倒置的目录树中，该目录树也称为文件系统层次结构。这棵树是倒置的，因为树根在该层次结构的顶部，树根的下方延伸出目录和子目录的分支。Linux 根目录（/）下包含很多的子目录（称为一级目录），同时，各一级目录下还含有很多子目录（称为二级目录）。

存储卷（volume）是对宿主机使用的本地文件存储系统、分布式文件存储系统等存储系统的抽象概念。存储卷的核心是一个目录，其中可能存有数据，应用实例可以访问该目录中的数据。存储卷的类型例如包括但不限于：用于数据缓存或临时存储的临时存储卷、网络文件系统（Network File System，NFS）。

持久卷（Persistent Volume，PV）是 K8s 集群中对存储资源的抽象，是存储卷的一种具体实现。PV 是 K8s 集群中定义的一种可被容器化应用使用的存储资源，PV 由管理员创建和配置，通过插件式的机制进行管理，供容器组（Pod）访问和使用。

Pod（容器组）：K8s 集群上处于运行状态的一组容器的集合，是应用实例的一种具体实现。

挂载点(mount point)：想要访问某个存储卷中的文件时，需要将其关联到宿主机目录树中的某个目录来实现，这种关联操作就是挂载（mount），这个目录就是挂载点，也即挂载点是文件在目录树中的位置。

在云计算场景中，如何通过存储卷让容器化应用使用宿主机的存储系统作为其数据共享和持久化的存储设施，是云计算领域面临的一大技术问题。为此，本申请实施例提供一种存储卷挂载与访问方法、设备及存储介质。在本申请实施例中，针对被应用实例绑定的存储卷，以子树挂载方式将存储卷挂载到宿主机目录树上，即在宿主机目录树上包括存储卷的根挂载点和指向根挂载点的子挂载点，使得应用实例能够通过宿主机目录树上的根挂载点访问存储卷，使得应用实例能够通过存储卷使用宿主机的存储系统作为数据共享和持

久化的存储设施的。

进一步，以子树挂载存储卷的方式，在存储卷被多个应用实例绑定的情况下，多个应用实例均通过根挂载点访问存储卷，能够共享相同的缓存空间，节约宿主机的内存资源。

进一步，子树挂载方式属于宿主机操作系统原生支持的挂载操作，在系统级的信息统计、挂载权限控制方面能够保持与独立挂载一样的逻辑和使用体验。例如，在同一存储卷被多个应用实例绑定的情况下，既能支持部分应用实例以读写方式对该存储卷进行挂载又能支持部分应用实例以只读方式对该存储卷进行挂载。又例如，操作系统可以感知同一个存储卷在宿主机目录树中的每一次挂载，为管理和维护存储卷的挂载提供了便利。

以下结合附图，详细说明本申请各实施例提供的技术方案。

图 1 为本申请实施例提供的一种存储卷挂载方法的流程图。参见图 1，该方法可以包括以下步骤：

100、确定被目标应用实例绑定且待挂载到宿主机目录树上的目标存储卷，宿主机目录树是目标应用实例所在宿主机的目录树。

101、判断宿主机目录树上是否包含目标存储卷的根挂载点；若判断结果为是，即宿主机目录树上包含目标存储卷的根挂载点，执行步骤 102；若判断结果为否，即宿主机目录树上未包含目标存储卷的根挂载点，执行步骤 103。

102、针对目标应用实例在宿主机目录树中创建指向根挂载点的子挂载点，执行步骤 104。

103、在宿主机目录树中创建目标存储卷的根挂载点，以及针对目标应用实例在宿主机目录树中创建指向根挂载点的子挂载点，执行步骤 104。

104、基于子挂载点以子树挂载方式将目标存储卷挂载到根挂载点上，以使目标应用实例通过根挂载点访问目标存储卷。

在本实施例中，目标应用实例承载于宿主机上，宿主机上可以承载一个或多个应用实例，目标应用实例可以是承载于宿主机的任一应用实例。进一步，本实施例的目标应用实例可以是容器化的应用实例，例如可以是容器或容器组（Pod）。从目标应用实例实现的功能来看，目标应用实例可以是但不限于：负载均衡实例、数据库实例、云服务器实例等等。宿主机可以是任何能够承载目标应用实例的广义上的计算机设备，例如可以是笔记本电脑、手机等终端设备，也可以是传统服务器、云服务器、服务器集群等服务器设备。在本实施例中，宿主机挂载有物理存储系统，例如块存储设备、磁盘、网络附属存储（Network Attached Storage, NAS）等，并通过宿主机目录树对该物理存储系统进行管理，宿主机目录树是宿主机上用于对宿主机的物理存储系统进行管理的目录树。进一步，NAS 可以包括 NFS（Network File System，网络文件系统）。

在本实施例中，为了便于对应用实例产生的数据进行共享和持久化，采用存储卷的概念，存储卷是定义在宿主机挂载的物理存储系统之上的存储空间，存储卷独立于应用实例的生命周期，因此它存储的数据可以在应用实例重启或重建后继续使用。在本实施例中，并不限定存储卷的实现方式，例如可以是PV，但并不限于此。用户可以按需预先声明目标应用实例需要绑定哪个或哪些存储卷，根据用户的预先声明可以为目标应用实例绑定一个或多个存储卷。其中，任意一个存储卷可以被宿主机上的一个或多个应用实例绑定。为了便于描述，将目标应用实例绑定的任一存储卷称为目标存储卷，并以目标存储卷为例对本申请实施例提供的存储卷挂载过程进行说明。为了便于目标应用实例通过存储卷使用宿主机的物理存储系统作为数据共享和持久化的存储设施，可以将目标应用实例绑定的目标存储卷挂载到宿主机目录树上（可简称为对目标存储卷进行挂载），即将存储卷与宿主机目录树上的某个目录建立关联关系。这样，当目标应用实例在目标存储卷中写入数据时，数据会被写入到宿主机上与该目标存储卷建立了关联关系的物理存储系统的相应目录下，实现数据的持久化存储。进一步，当其它应用实例与目标应用实例绑定同一目标存储卷时，还可以在其它应用实例与目标应用实例之间实现数据的共享。

在本实施例中，对目标存储卷进行挂载时，与传统的存储卷挂载方式不同，既不是采用独立挂载方式，也不是采用挂载根目录的方式，而是采用子树挂载方式将目标存储卷挂载到宿主机目录树上。

其中，独立挂载方式是指对于同一存储卷，在该存储卷被不同应用实例绑定的情况下，针对不同应用实例对该存储卷进行挂载都是独立的，从宿主机操作系统角度来看，不同应用实例对同一存储卷的挂载会形成一颗独立的目录树，且彼此之间是完全隔离且不共享任何资源的。其中，在独立挂载方式下，同一存储卷的缓存数据需要针对不同应用实例存储多份，会浪费宿主机的缓存资源。

其中，挂载根目录的方式是指对于同一存储卷，在首次挂载时将其挂载到宿主机目录树的根目录下，对于后续绑定该存储卷的其它应用实例以链接指向的方式直接复用该唯一的挂载点。在挂载根目录的方式下，因为不同应用实例复用同一个挂载点，不同应用实例只能使用相同的挂载权限，或者是只读，或者是读写，在挂载权限控制方面不够灵活；另外，从宿主机操作系统的角度来看，这种方式相当于只挂载了一次，所以操作系统无法了解到针对不同应用实例对该存储卷的挂载信息，会对存储卷的管理和维护造成不便。

本实施例采用的子树挂载方式是指对于同一存储卷，以目标存储卷为例，在宿主机目录树上为该目标存储卷创建根挂载点，另外，针对绑定该目标存储卷的不同应用实例，在宿主机目录树上为该目标存储卷分别创建指向根挂载点的子挂载点，以便通过根挂载点访问目标存储卷。在子树挂载方式中，在宿主机目录树上会出现目标存储卷的一个根挂载点

和指向根挂载点的至少一个子挂载点，根挂载点和至少一个子挂载点形成该目标存储卷对应的一颗挂载目录树，并以挂载目录树为粒度为目标存储卷进行资源分配和管理。其中，子挂载点的数量根据绑定目标存储卷的应用实例的数量而定。

在子树挂载方式中，对不同应用实例而言，在访问目标存储卷时都要通过根挂载点，
5 通过根挂载点能够在绑定同一目标存储卷的不同应用实例之间建立关联关系，实现资源共享，目标存储卷的缓存数据只需存储一份，有利于节约宿主机的缓存资源。另外，对绑定同一目标存储卷的不同应用实例来说，又具有各自的子挂载点，而不是完全复用同一挂载点，允许不同应用实例采用不同的挂载权限进行挂载，即允许一些应用实例以只读方式挂载目标存储卷，允许一些应用实例以读写方式挂载目标存储卷，在挂载权限控制方面具有
10 较高的灵活性。

在本实施例中，对目标存储卷进行挂载的挂载时机不做限制，可以是目标应用实例访问目标存储卷之前的任何时机。其中，在宿主机所在的集群中，存在对各应用实例进行管理的主节点（Master），该主节点可以监测计算集群中是否有新的应用实例被创建，应用实例是否绑定了存储卷等信息，以负责将应用实例调度到集群中的某个计算节点上，应用实例被调度到的计算节点即为承载该应用实例的宿主机。在该场景中，主节点还可以向承载应用实例的宿主机发送挂载指令，以指示宿主机为应用实例绑定的存储卷执行挂载操作。基于前述，将目标存储卷挂载到宿主机目录树上的挂载时机包括但不限于以下几种情况：

例如，可以在接收到与宿主机处于同一计算集群中具有管理功能的主节点发送的挂载指令时，执行对目标存储卷进行挂载的挂载操作；其中，挂载指令指示宿主机对目标应用
20 实例绑定的目标存储卷进行挂载操作。

又例如，在监测到目标应用实例被调度到宿主机上时，自动为目标应用实例进行宿主机目录树的建立与持久化，并执行对目标存储卷进行挂载的挂载操作。其中，宿主机目录树的持久化是指将宿主机目录树与物理存储系统进行关联的过程。

又例如，在监测到目标应用实例在宿主机上被创建，但目标应用实例绑定的目标存储卷尚未挂载到宿主机目录树上时，则执行对目标存储卷进行挂载的挂载操作。
25

又例如，在目标应用实例向目标存储卷发起数据访问请求时，若发现目标存储卷尚未挂载到宿主机目录树上，则执行对目标存储卷进行挂载的挂载操作。

在本实施例中，在将目标存储卷挂载到宿主机目录树上时，可以调用宿主机操作系统支持的挂载命令，由宿主机的操作系统执行挂载操作，但并不限于此。例如，也可以在宿主机上部署一存储卷挂载进程，由该存储卷挂载进程执行本实施例提供的存储卷挂载操作。
30 其中，无论执行主体是操作系统还是存储卷挂载进程，在执行挂载操作时，首先确定宿主机目录树上是否包含目标存储卷的根挂载点。其中，目标存储卷在宿主机目录树上会关联

的一个与绑定该目标存储卷的应用实例无关的目录，将该目录称为目标存储卷在宿主机目录树上的根挂载点。其中，作为根挂载点的目录可以由用户预先声明、指定或配置。

可选的，为了快速准确地确定宿主机目录树上是否包含目标存储卷的根挂载点，可以为宿主机目录树维护挂载列表，并将挂载到宿主机目录树上的存储卷的卷标识及其根挂载点关联记录在挂载列表中，即挂载列表中记录有已挂在到宿主机目录树上的存储卷与根挂载点之间的映射关系。于是，确定宿主机目录树上是否包含目标存储卷的根挂载点时，可以根据目标存储卷的标识在宿主机目录树对应的挂载列表中进行匹配；若匹配到目标存储卷的标识，将与目标存储卷的标识对应的根挂载点作为目标存储卷的根挂载点；若未匹配到目标存储卷的标识，确定宿主机目录树上未包含目标存储卷的根挂载点。

10 在本实施例中，针对目标存储卷的每一次挂载，可在宿主机目录树上针对目标存储卷为本次挂载创建子挂载点，子挂载点是根挂载点的子节点，根挂载点是子挂载点的父节点，进而实现以子树挂载方式将目标存储卷挂载到根挂载点上。实际应用中，目标存储卷可以被一个或多个应用实例绑定，当目标存储卷被一个应用实例绑定时，在宿主机目录树上会出现目标存储卷的一个根挂载点和指向根挂载点的一个子挂载点，此时该目标存储卷对应一颗具有一个子挂载点的挂载目录树；随着时间推移，当目标存储卷被多个应用实例绑定时，在宿主机目录树上会出现目标存储卷的一个根挂载点和指向根挂载点的多个子挂载点，此时该目标存储卷对应的一颗具有多个子挂载点的挂载目录树。

基于上述，若宿主机目录树上包含目标存储卷的根挂载点，则可以直接针对目标应用实例在宿主机目录树中创建指向根挂载点的子挂载点；然后，基于子挂载点以子树挂载方式将目标存储卷挂载到根挂载点上，以使目标应用实例通过根挂载点访问目标存储卷。若宿主机目录树上未包含目标存储卷的根挂载点，在创建子挂载点之前，在宿主机目录树中创建目标存储卷的根挂载点；接着，针对目标应用实例在宿主机目录树中创建指向根挂载点的子挂载点；然后，基于子挂载点以子树挂载方式将目标存储卷挂载到根挂载点上，以使目标应用实例通过根挂载点访问目标存储卷。

25 可选的，用户可以通过配置文件的方式，预先配置目标应用实例在宿主机目录树中的目录位置。基于此，针对目标应用实例在宿主机目录树中创建指向根挂载点的子挂载点时，可以从目标应用实例对应的配置文件中，获取目标应用实例在宿主机目录树中对应的目录位置；将目录位置作为子挂载点，并建立子挂载点与根挂载点之间的指向关系。示例性的，建立子挂载点与根挂载点之间的指向关系时，可以将根挂载点作为子挂载点的父挂载点，并将根挂载点的标识记录子挂载点的属性信息中。

30 在本实施例中，子挂载点可以理解为根挂载点的下一级目录，根挂载点可以理解为子挂载点的上一级目录。子挂载点与根挂载点之间存在指向关系，也即子挂载点与根挂载点

绑定，当子挂载点与根挂载点之间不存在指向关系时，子挂载点与根挂载点解除绑定。

可选的，为了便于维护和管理存储卷的挂载操作，在基于子挂载点以子树挂载方式将目标存储卷挂载到根挂载点上之后，还可以根据设定的次数更新步长，增加根挂载点的引用次数，其中，次数更新步长为正整数。次数更新步长根据实际应用需求灵活设置，例如
5 为 1、2、3 等等。根挂载点的引用次数反映目标存储卷的挂载次数。以次数更新步长为 1 为例，目标存储卷每挂载一次，对应的根挂载点的引用次数增加 1。

可选的，为了便于维护和管理存储卷的挂载操作，在目标应用实例从宿主机上销毁的情况下，根据次数更新步长，还可以减少根挂载点的引用次数；以及在根挂载点的引用次数为预设初始值的情况下，从宿主机目录树上卸载根挂载点。

10 具体而言，在目标应用实例从宿主机上销毁时，对应的子挂载点也会被销毁，也即对应的子挂载点从宿主机目录树上删除，对应的子挂载点与根挂载点解除绑定。另外，在目标应用实例从宿主机上销毁时，还需根据次数更新步长，减少根挂载点的引用次数。以次数更新步长为 1 为例，目标应用实例被销毁时，对应的根挂载点的引用次数减少 1，随着
15 绑定目标存储卷的应用实例的不断被销毁，根挂载点的引用次数会不断被减少直至为预设初始值。若根挂载点的引用次数减少到预设初始值，则从宿主机目录树上卸载根挂载点，也即宿主机目录树上不再存在目标存储卷的根挂载点。其中，预设初始值按需灵活设置，例如为 0、1、2 等等。

在本申请实施例中，针对被应用实例绑定的存储卷，以子树挂载方式将存储卷挂载到宿主机目录树上，即在宿主机目录树上包括存储卷的根挂载点和指向根挂载点的子挂载点，
20 使得应用实例能够通过宿主机目录树上的根挂载点访问存储卷，进而达到通过存储卷使用宿主机的存储系统作为应用实例实现数据共享和持久化的存储设施的目的。

进一步，以子树挂载存储卷的方式，在存储卷被多个应用实例绑定的情况下，多个应用实例均通过根挂载点访问存储卷，多个应用实例之间能够通过根挂载点共享相同的缓存空间，节约宿主机的内存资源。

25 进一步，子树挂载方式属于宿主机操作系统原生支持的挂载操作，在系统级的信息统计、挂载权限控制方面能够保持与独立挂载一样的逻辑和使用体验。例如，能够继续支持读写、只读等挂载权限控制，针对被多个应用实例绑定的同一个存储卷，既支持部分应用实例以读写方式对该存储卷进行挂载又支持部分应用实例以只读方式对该存储卷进行挂载。又例如，基于子挂载点的创建，操作系统可以感知同一个存储卷在宿主机目录树中的
30 每一次挂载，为管理和维护存储卷的挂载提供了便利，便于针对存储卷进行系统级信息的统计，具有更好的用户体验。

图 2 为本申请实施例提供的另一种存储卷挂载方法的流程图。参见图 2，该方法可以

包括以下步骤：

200、确定被目标应用实例绑定且待挂载到宿主机目录树上的目标存储卷，宿主机目录树是目标应用实例所在宿主机的目录树。

201、判断宿主机目录树上是否包含目标存储卷的根挂载点；若判断结果为否，即宿主机目录树上未包含目标存储卷的根挂载点，执行步骤 202、203 和 204；若判断结果为是，即宿主机目录树上包含目标存储卷的根挂载点，执行步骤 203 和 204。

202、在宿主机目录树中创建目标存储卷的根挂载点，以及根据目标存储卷的根挂载点为目标存储卷分配缓存空间，并将目标存储卷的缓存空间与目标存储卷的根挂载点进行绑定，并执行步骤 203。

203、针对目标应用实例在宿主机目录树中创建指向根挂载点的子挂载点，并执行步骤 204。

204、基于子挂载点以子树挂载方式将目标存储卷挂载到根挂载点上，以使目标应用实例通过根挂载点优先访问目标存储卷的缓存空间。

本实施例中的步骤 200、步骤 201、步骤 203 和步骤 204 的实现方式可以参见前述实施例的相关介绍，在此不再赘述。

在本实施例中，在为目标存储卷创建根挂载点的情况下，还可以根据根挂载点为目标存储卷分配缓存空间，并将目标存储卷的缓存空间与目标存储卷的根挂载点进行绑定，该缓存空间用于存储目标存储卷对应的缓存数据，例如目标存储卷中的数据、元数据、目录项等。需要说明的是，缓存数据会随着目标存储卷的访问情况动态变化。在本申请实施例中，以挂载目录树为粒度为目标存储卷分配缓存资源，对于绑定目标存储卷的各个应用实例均可以通过子挂载点与根挂载点的指向关系共享该缓存空间，而不用为每个应用实例都维护一份缓存数据，有利于节约宿主机的缓存资源。

在将目标存储卷挂载到宿主机目录树上之后，绑定目标存储卷的目标应用实例就可以通过宿主机目录树对目标存储卷进行访问。在以子树挂载方式将目标存储卷挂载到宿主机目录树的基础上，本申请实施例还提供了存储卷访问的方法。

图 3 为本申请实施例提供的一种存储卷访问方法的流程图。参见图 3，该方法可以包括以下步骤：

301、响应目标应用实例对目标存储卷的访问操作，根据目标存储卷在宿主机目录树上的根挂载点和子挂载点，生成目标存储卷的访问路径。

302、根据访问路径，对目标存储卷执行相应类型的访问操作，其中，目标存储卷是被目标应用实例绑定且挂载到宿主机目录树上的存储卷。

在本实施例中，目标存储卷是被目标应用实例绑定且挂载到宿主机目录树上的存储卷，

且目标存储卷是以子树挂载方式将被挂载到宿主机目录树上的，关于挂载过程的相关描述，可参见前述实施例，在此不再赘述。

在将目标存储卷挂载到宿主机目录树上之后，目标应用实例可以对目标存储卷发起访问操作。访问操作例如包括但不限于：打开目标存储卷中的文件、读/写目标存储卷中的文件、关闭目标存储卷中的文件等。目标应用实例可以通过可移植操作系统接口（Portable Operating System Interface, POSIX）接口对目标存储卷发起访问操作，但并不以此为限。其中，对目标存储卷的访问操作可由宿主机的操作系统执行，对此也不做限定。

在目标应用实例对目标存储卷发起访问操作后，可以查询宿主机目录树，确定目标存储卷在宿主机目录树上的根挂载点和子挂载点，这里的子挂载点是指目标存储卷针对发起访问操作的目标应用实例在宿主机目录树上的子挂载点。接着，根据目标存储卷在宿主机目录树上的根挂载点和子挂载点，生成目标存储卷的访问路径；根据访问路径，对目标存储卷执行相应类型的访问操作。访问操作的类型例如包括但不限于：打开、读/写、关闭文件等。需要说明的是，随着访问操作的执行，被访问文件的最近访问时间、访问对象等元数据会发生变化。

在本实施例中，子挂载点可以理解为根挂载点的下一级目录，根挂载点可以理解为子挂载点的上一级目录。在根据目标存储卷在宿主机目录树上的根挂载点和子挂载点，生成目标存储卷的访问路径时，可以根据根挂载点生成主访问路径，根据子挂载点生成子访问路径；然后，根据子挂载点与根挂载点之间存在指向关系，将主访问路径与子访问路径进行拼接，以得到目标存储卷的访问路径。

图 4 为本申请实施例提供的另一种存储卷访问方法的流程图。参见图 4，该方法可以包括以下步骤：

401、响应目标应用实例对目标存储卷的访问操作，确定与目标存储卷的根挂载点绑定的目标存储卷的缓存空间。

402、访问目标存储卷的缓存空间，其中，目标存储卷是被目标应用实例绑定且挂载到宿主机目录树上的存储卷。

403、若目标存储卷的缓存空间未包含访问操作所需的数据，根据目标存储卷在宿主机目录树上的根挂载点和子挂载点，生成目标存储卷的访问路径。

404、根据访问路径，对目标存储卷执行相应类型的访问操作。

在本实施例中，在目标应用实例对目标存储卷发起访问操作后，可以优先访问目标存储卷的缓存空间，并在目标存储卷的缓存空间未包含访问操作所需的数据的情况下，根据目标存储卷在宿主机目录树上的根挂载点和子挂载点，生成目标存储卷的访问路径，根据该访问路径访问目标存储卷。进一步，在目标存储卷的缓存空间中存在访问操作所需的数

据的情况下，可以直接访问宿主机本地的缓存空间，而无需访问远程的物理存储系统，有利于提高数据访问速度和效率。另外，关于本实施例中其它步骤的描述可参见前述实施例，在此不再赘述。

本申请实施例提供的技术方案，针对被应用实例绑定的存储卷，以子树挂载方式挂载到宿主机目录树，以及将目标存储卷的缓存空间与目标存储卷的根挂载点进行绑定，使得多个应用实例之间可以共享目标存储卷的缓存空间，无需为同一个存储卷配置多个缓存空间，节约了宿主机的缓存资源。这样，应用实例在对存储卷发起访问操作时，可以优先访问目标存储卷的缓存空间，并在目标存储卷的缓存空间未包含访问操作所需的数据的情况下再访问目标存储卷，有利于提高数据访问效率和速度。

为了更好地理解本申请实施例提供的技术方案，下面以 K8s (Kubernetes 的简称) 集群场景为例，对本申请实施例提供的技术方案进行详细说明。

在 K8s 集群中，提出了 PV 的概念，且允许 Pod 绑定 PV，且允许多个 Pod 绑定同一 PV，通过将 PV 挂载到宿主机的目录树上，使得 Pod 能够通过宿主机的存储系统实现数据共享和持久化存储。在传统方案中，每个 Pod 对 PV 的挂载都是独立的，从宿主机的操作系统的角度来看，每次挂载都是一颗独立的目录树，这些目录树之间是完全隔离且不共享任何资源。这种独立挂载方式，在不同 Pod 绑定相同 PV 时，不同 Pod 对相同 PV 的缓存资源是不共享的，需要为不同 Pod 分别维护一份 PV 的缓存数据，这会浪费宿主机的缓存资源。

鉴于上述问题，采用本申请实施例可以针对 PV 提供一种新的挂载方案，即以子树挂载方式对 PV 进行挂载。具体地，参见图 5，以 K8s 集群中的一个工作节点 (Node) 为例，该 Node 上可以部署一个或多个 Pod，在图 5 中，以 Node 上部署 3 个不同的 Pod，即 Pod1、Pod2 和 Pod3 为例，且 3 个 Pod 需要绑定通过一下 PV，假设 3 个 Pod 共同绑定的持久卷记为 PV1。

在 Pod1 访问 PV1 之前，首先要按照下述步骤完成对 PV1 的挂载：根据 PV1 的名称查询操作系统维护的挂载列表中是否存在 PV1 的根挂载点，若挂载列表中不存在 PV1 的根挂载点，在宿主机目录树上创建 PV1 的根挂载点，一方面将 PV1 的根挂载点以及 PV1 的名称关联存储至挂载列表中，另一方面为 PV1 在宿主机目录树上确定一个指向 PV1 的根挂载点的子挂载点 1，将 PV1 挂载到宿主机目录树的子挂载点 1 上，并将 PV1 的根挂载点与一个缓存空间绑定。若挂载列表中存在 PV1 的根挂载点，直接为 PV1 在宿主机目录树上确定一个指向 PV1 的根挂载点的子挂载点 1，将 PV1 挂载到宿主机目录树的子挂载点 1 上。同理，在 Pod2 或 Pod3 访问 PV1 之前，需要按照上述流程以子树挂载方式完成对 PV1 的挂载。

在图 5 中, 以 PV1 的根挂载点为一级目录为例进行图示, 且针对 Pod1, PV1 的子挂载点为某个二级目录; 针对 Pod2, PV1 的子挂载点为另一二级目录; 针对 Pod3, PV1 的子挂载点为某个三级目录为例进行图示, 三个子挂载点均指向根挂载点。

无论是哪个 Pod, 当针对该 Pod 完成对 PV1 的挂载之后, 在该 Pod 对 PV1 发起访问操作时, 5 优先访问与 PV1 的根挂载点绑定的缓存空间, 并在 PV1 的缓存空间未包含访问操作所需的数据的情况下再根据根挂载点和对应的子挂载点去访问 PV1。

值得注意的是, 图 5 中的宿主机目录树是工作节点挂载的文件存储系统的目录树。

其中, 针对 PV1 的每一次挂载操作, 因为需要为其创建子挂载点, 所以操作系统都可以感知, 这样方便操作系统针对 PV1 进行系统级信息统计, 例如统计针对 PV1 进行了几次挂载, 每次针对 PV1 发起挂载的 Pod 是哪个等信息。进一步, 在工作节点上可以部署一守护进程, 并在针对 PV1 执行首次挂载操作时, 启动该守护进程, 由该守护进程监测针对 PV1 的挂载操作, 当监测到针对 PV1 创建根挂载点时, 初始化该根挂载点的引用次数为 0, 并在监测到针对 PV1 每增加一个指向该根挂载点的子挂载点时, 将根挂载点的引用次数加 1。另外, 当监测到绑定 PV1 的某个 Pod (例如 Pod1) 从 Node 上销毁时, 将与被销毁的 Pod 对应的子挂载点从宿主机目录树上删除, 并将根挂载点的引用次数减 1。若守护进程监测到根挂载点的引用次数重新变为 0 时, 提示操作系统可以从宿主机目录树上卸载该根挂载点。

在本实施例中, 以子树挂载方式针对 Pod 绑定的 PV 进行挂载, 首先, 使得 Pod 能够通过宿主机目录树上的根挂载点访问 PV, 使得 Pod 能够通过 PV 使用宿主机的存储系统作为数据共享和持久化的存储设施。进一步地, 在多个 Pod 绑定同一 PV 的情况下, 以子树挂载 PV 的方式, 通过根挂载点可以让多个 Pod 共享相同的缓存资源 (包括数据、元数据和目录项等), 可节约宿主机的内存资源, 提升宿主机的资源利用率。另外, 以子树挂载 PV 属于操作系统原生支持的挂载操作, 借助子树挂载, 针对同一个 PV 在被多个 Pod 绑定的情况下, 既支持部分 Pod 对其以读写方式进行挂载, 又支持部分 Pod 对其以只读方式进行挂载, 在挂载权限控制方面具有较好的灵活性。再者, 以子树挂载 PV, 针对 PV 的每次挂载操作, 操作系统都可以感知, 因此可以针对 PV 进行各种系统级信息的统计, 例如 PV 被挂载的次数, 被哪些 Pod 绑定了, 以及每次挂载对磁盘空间的占用情况等, 为管理和维护 PV 的挂载提供了便利。

图 6 为本申请实施例提供的一种存储卷挂载装置的结构示意图。参见图 6, 该装置可以包括: 确定模块 61, 用于确定被目标应用实例绑定且待挂载到宿主机目录树上的目标存储卷, 宿主机目录树是目标应用实例所在宿主机的目录树; 创建模块 62, 用于若宿主机目录树上包含目标存储卷的根挂载点, 针对目标应用实例在宿主机目录树中创建指向根挂载

点的子挂载点；挂载模块 63，用于基于子挂载点以子树挂载方式将目标存储卷挂载到根挂载点上，以使目标应用实例通过根挂载点访问目标存储卷。

进一步可选的，创建模块 62 还用于若宿主机目录树上未包含目标存储卷的根挂载点，在创建子挂载点之前，在宿主机目录树中创建目标存储卷的根挂载点。

5 进一步可选的，目标存储卷能够被宿主机上的多个应用实例绑定，且多个应用实例包括目标应用实例。

进一步可选的，上述装置还包括匹配模块，用于：根据目标存储卷的标识在宿主机目录树对应的挂载列表中进行匹配，挂载列表中记录有已挂在到宿主机目录树上的存储卷与根挂载点之间的映射关系；若匹配到目标存储卷的标识，将与目标存储卷的标识对应的根
10 挂载点作为目标存储卷的根挂载点；若未匹配到目标存储卷的标识，确定宿主机目录树上未包含目标存储卷的根挂载点。

进一步可选的，创建模块 62 针对目标应用实例在宿主机目录树中创建指向根挂载点的子挂载点时，具体用于：从目标应用实例对应的配置文件中，获取目标应用实例在宿主机目录树中对应的目录位置；将目录位置作为子挂载点，并建立子挂载点与根挂载点之间的
15 指向关系。

进一步可选的，创建模块 62 建立子挂载点与根挂载点之间的指向关系时，具体用于：将根挂载点作为子挂载点的父挂载点，并将根挂载点的标识记录于子挂载点的属性信息中。

进一步可选的，挂载模块 63 还用于根据设定的次数更新步长，增加根挂载点的引用次数，其中，次数更新步长为正整数。

20 进一步可选的，挂载模块 63 还用于在目标应用实例从宿主机上销毁的情况下，根据次数更新步长，减少根挂载点的引用次数；以及在根挂载点的引用次数为预设初始值的情况下，从宿主机目录树上卸载根挂载点。

进一步可选的，上述装置还包括访问模块，用于响应目标应用实例对目标存储卷的访问操作，根据目标存储卷在宿主机目录树上的根挂载点和子挂载点，生成目标存储卷的访问
25 路径；根据访问路径，对目标存储卷执行相应类型的访问操作。

进一步可选的，创建模块 62 还用于根据目标存储卷的根挂载点为目标存储卷分配缓存空间，并将目标存储卷的缓存空间与目标存储卷的根挂载点进行绑定。基于此，访问模块具体用于：响应目标应用实例对目标存储卷的访问操作，确定与目标存储卷的根挂载点绑定的目标存储卷的缓存空间；访问目标存储卷的缓存空间；若目标存储卷的缓存空间未
30 包含访问操作所需的数据，根据目标存储卷在宿主机目录树上的根挂载点和子挂载点，生成目标存储卷的访问路径；根据访问路径，对目标存储卷执行相应类型的访问操作。

图 6 所示的装置可以执行图 1 至图 3 所示的方法，其实现原理和技术效果不再赘述。

对于上述实施例中的图 6 所示的装置其中各个模块、单元执行操作的具体方式已经在有关该方法的实施例中进行了详细描述，此处将不做详细阐述说明。

图 7 为本申请实施例提供的一种存储卷访问装置的结构示意图。参见图 7，该装置可以包括：生成模块 71，用于响应目标应用实例对目标存储卷的访问操作，根据目标存储卷在宿主机目录树上的根挂载点和子挂载点，生成目标存储卷的访问路径；访问模块 72，用于根据访问路径，对目标存储卷执行相应类型的访问操作，其中，目标存储卷是被目标应用实例绑定且挂载到宿主机目录树上的存储卷。

进一步可选的，上述装置还包括：确定模块，用于在生成模块 71 生成访问路径之前，确定与目标存储卷的根挂载点绑定的目标存储卷的缓存空间。基于此，访问模块 72 还用于：访问目标存储卷的缓存空间；若目标存储卷的缓存空间未包含访问操作所需的数据，则执行生成目标存储卷的访问路径的步骤；在目标存储卷的缓存空间中存在访问操作所需的数据的情况下，直接从目标存储卷的缓存空间中获取访问操作所需的数据。

图 7 所示的装置可以执行图 4 所示的方法，其实现原理和技术效果不再赘述。对于上述实施例中的图 7 所示的装置其中各个模块、单元执行操作的具体方式已经在有关该方法的实施例中进行了详细描述，此处将不做详细阐述说明。

需要说明的是，在上述实施例及附图中的描述的一些流程中，包含了按照特定顺序出现的多个操作，但是应该清楚了解，这些操作可以不按照其在本文中出现的顺序来执行或并行执行，操作的序号如 401、402 等，仅仅是用于区分开各个不同的操作，序号本身不代表任何的执行顺序。另外，这些流程可以包括更多或更少的操作，并且这些操作可以按顺序执行或并行执行。需要说明的是，本文中的“第一”、“第二”等描述，是用于区分不同的消息、设备、模块等，不代表先后顺序，也不限定“第一”和“第二”是不同的类型。

需要说明的是，本申请所涉及的用户信息（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。

图 8 为本申请实施例提供的一种计算机设备的结构示意图。如图 8 所示，该计算机设备包括：存储器 81 和处理器 82。

存储器 81，用于存储计算机程序，并可被配置为存储其它各种数据以支持在计算机设备上的操作。这些数据的示例包括用于在计算机设备上操作的任何应用程序或方法的指令，消息，图片，视频等。

存储器 81 可以由任何类型的易失性或非易失性存储设备或者它们的组合实现，如静

态随机存取存储器 (Static Random-AccessMemory, SRAM), 电可擦除可编程只读存储器 (Electrically Erasable Programmable read only memory, EEPROM), 可擦除可编程只读存储器 (Erasable Programmable Read Only Memory, EPROM), 可编程只读存储器 (Programmable read-only memory, PROM), 只读存储器 (Read-Only Memory, ROM), 磁存储器, 快闪存
5 存储器, 磁盘或光盘。

处理器 82, 与存储器 81 耦合, 用于执行存储器 81 中的计算机程序, 以用于实现本申请实施例提供的存储卷挂载方法和/或存储卷访问方法中的步骤。

在一些实施例中, 处理器 82 执行存储器 81 中的计算机程序, 以用于: 确定被目标应用实例绑定且待挂载到宿主机目录树上的目标存储卷, 宿主机目录树是目标应用实例所在
10 宿主机的目录树; 判断宿主机目录树上是否包含目标存储卷的根挂载点; 若判断结果为是, 即宿主机目录树上包含目标存储卷的根挂载点, 针对目标应用实例在宿主机目录树中创建指向根挂载点的子挂载点, 基于子挂载点以子树挂载方式将目标存储卷挂载到根挂载点上, 以使目标应用实例通过根挂载点访问目标存储卷。

进一步可选地, 若判断结果为否, 即宿主机目录树上未包含目标存储卷的根挂载点, 在宿主机目录树中创建目标存储卷的根挂载点, 以及针对目标应用实例在宿主机目录树中
15 创建指向根挂载点的子挂载点, 并基于子挂载点以子树挂载方式将目标存储卷挂载到根挂载点上, 以使目标应用实例通过根挂载点访问目标存储卷。

在一可选实施例中, 目标存储卷能够被宿主机上的多个应用实例绑定, 且多个应用实例包括目标应用实例。

20 在一可选实施例中, 处理器 82 还用于: 根据目标存储卷的标识在宿主机目录树对应的挂载列表中进行匹配, 挂载列表中记录有已挂在到宿主机目录树上的存储卷与根挂载点之间的映射关系; 若匹配到目标存储卷的标识, 将与目标存储卷的标识对应的根挂载点作为目标存储卷的根挂载点; 若未匹配到目标存储卷的标识, 确定宿主机目录树上未包含目标存储卷的根挂载点。

25 在一可选实施例中, 处理器 82 在针对目标应用实例在宿主机目录树中创建指向根挂载点的子挂载点时, 具体用于: 从目标应用实例对应的配置文件中, 获取目标应用实例在宿主机目录树中对应的目录位置; 将目录位置作为子挂载点, 并建立子挂载点与根挂载点之间的指向关系。

30 在一可选实施例中, 处理器 82 在建立子挂载点与根挂载点之间的指向关系时, 具体用于: 将根挂载点作为子挂载点的父挂载点, 并将根挂载点的标识记录子挂载点的属性信息中。

在一可选实施例中, 处理器 82 在基于子挂载点以子树挂载方式将目标存储卷挂载到

根挂载点上之后，还用于：根据设定的次数更新步长，增加根挂载点的引用次数，其中，次数更新步长为正整数。

在一可选实施例中，处理器 82 还用于：在目标应用实例从宿主机上销毁的情况下，根据次数更新步长，减少根挂载点的引用次数；以及在根挂载点的引用次数为预设初始值的情况下，从宿主机目录树上卸载根挂载点。

在一可选实施例中，处理器 82 还用于：响应目标应用实例对目标存储卷的访问操作，根据目标存储卷在宿主机目录树上的根挂载点和子挂载点，生成目标存储卷的访问路径；根据访问路径，对目标存储卷执行相应类型的访问操作。

在一可选实施例中，处理器 82 还用于：根据目标存储卷的根挂载点为目标存储卷分配缓存空间，并将目标存储卷的缓存空间与目标存储卷的根挂载点进行绑定。

在一可选实施例中，处理器 82 具体用于：响应目标应用实例对目标存储卷的访问操作，确定与目标存储卷的根挂载点绑定的目标存储卷的缓存空间；若目标存储卷的缓存空间未包含访问操作所需的数据，根据目标存储卷在宿主机目录树上的根挂载点和子挂载点，生成目标存储卷的访问路径；根据访问路径，对目标存储卷执行相应类型的访问操作。

在一些实施例中，处理器 82 执行存储器 81 中的计算机程序，以用于：响应目标应用实例对目标存储卷的访问操作，根据目标存储卷在宿主机目录树上的根挂载点和子挂载点，生成目标存储卷的访问路径；根据访问路径，对目标存储卷执行相应类型的访问操作，其中，目标存储卷是被目标应用实例绑定且挂载到宿主机目录树上的存储卷。

在一可选实施例中，处理器 82 在生成目标存储卷的访问路径之前，还用于：确定与目标存储卷的根挂载点绑定的目标存储卷的缓存空间；若目标存储卷的缓存空间未包含访问操作所需的数据，则执行生成目标存储卷的访问路径的步骤。

关于各方法中的步骤请参见前述实施例，在此不再赘述。

进一步，如图 8 所示，该计算机设备还包括：通信组件 83、显示器 84、电源组件 85、音频组件 86 等其它组件。图 8 中仅示意性给出部分组件，并不意味着计算机设备只包括图 8 所示组件。另外，图 8 中虚线框内的组件为可选组件，而非必选组件，具体可视计算机设备的产品形态而定。本实施例的计算机设备可以实现为台式电脑、笔记本电脑、智能手机或 IOT（物联网，Internet of things）设备等终端设备，也可以是常规服务器、云服务器或服务器阵列等服务端设备。若本实施例的计算机设备实现为台式电脑、笔记本电脑、智能手机等终端设备，可以包含图 8 中虚线框内的组件；若本实施例的计算机设备实现为常规服务器、云服务器或服务器阵列等服务端设备，则可以不包含图 8 中虚线框内的组件。

关于处理器执行各动作的详细实施过程可参见前述方法实施例或设备实施例中的相关描述，在此不再赘述。

相应地，本申请实施例还提供一种存储有计算机程序的计算机可读存储介质，计算机程序被执行时能够实现本申请实施例提供的存储卷挂载方法和/或存储卷访问方法中的步骤。

相应地，本申请实施例还提供一种计算机程序产品，包括计算机程序/指令，当计算机程序/指令被处理器执行时，致使处理器能够实现本申请实施例提供的存储卷挂载方法和/或存储卷访问方法中的步骤。

上述通信组件被配置为便于通信组件所在设备和其他设备之间有线或无线方式的通信。通信组件所在设备可以接入基于通信标准的无线网络，如 WiFi（Wireless Fidelity，无线保真）、2G（2Generation，2 代）、3G（3Generation，3 代）、4G（4Generation，4 代）/LTE（long Term Evolution，长期演进）、5G（5Generation，5 代）等移动通信网络，或它们的组合。在一个示例性实施例中，通信组件经由广播信道接收来自外部广播管理系统的广播信号或广播相关信息。在一个示例性实施例中，所述通信组件还包括近场通信（Near Field Communication，NFC）模块，以促进短程通信。例如，在 NFC 模块可基于射频识别（Radio Frequency Identification，RFID）技术，红外数据协会（The Infrared Data Association，IrDA）技术，超宽带（Ultra Wide Band，UWB）技术，蓝牙（Bluetooth，BT）技术和其他技术来实现。

上述显示器包括屏幕，其屏幕可以包括液晶显示器（Liquid Crystal Display，LCD）和触摸面板（Touch Panel，TP）。如果屏幕包括触摸面板，屏幕可以被实现为触摸屏，以接收来自用户的输入信号。触摸面板包括一个或多个触摸传感器以感测触摸、滑动和触摸面板上的手势。所述触摸传感器可以不仅感测触摸或滑动动作的边界，而且还检测与所述触摸或滑动操作相关的持续时间和压力。

上述电源组件，为电源组件所在设备的各种组件提供电力。电源组件可以包括电源管理系统，一个或多个电源，及其他与为电源组件所在设备生成、管理和分配电力相关联的组件。

上述音频组件，可被配置为输出和/或输入音频信号。例如，音频组件包括一个麦克风（microphone，MIC），当音频组件所在设备处于操作模式，如呼叫模式、记录模式和语音识别模式时，麦克风被配置为接收外部音频信号。所接收的音频信号可以被进一步存储在存储器或经由通信组件发送。在一些实施例中，音频组件还包括一个扬声器，用于输出音频信号。

本领域内的技术人员应明白，本申请的实施例可提供为方法、系统、或计算机程序产品。因此，本申请可采用完全硬件实施例、完全软件实施例、或结合软件和硬件方面的实施例的形式。而且，本申请可采用在一个或多个其中包含有计算机可用程序代码的计算机

可读存储介质（包括但不限于磁盘存储器、CD-ROM、光学存储器等）上实施的计算机程序产品的形式。

本申请是参照根据本申请实施例的方法、设备（系统）、和计算机程序产品的流程图和 / 或方框图来描述的。应理解可由计算机程序指令实现流程图和 / 或方框图中的每一流程
5 和 / 或方框、以及流程图和 / 或方框图中的流程和 / 或方框的结合。可提供这些计算机程序指令到通用计算机、专用计算机、嵌入式处理机或其他可编程数据处理设备的处理器以产生一个机器，使得通过计算机或其他可编程数据处理设备的处理器执行的指令产生用于实现在流程图一个流程或多个流程和 / 或方框图一个方框或多个方框中指定的功能的装置。

10 这些计算机程序指令也可存储在能引导计算机或其他可编程数据处理设备以特定方式工作的计算机可读存储器中，使得存储在该计算机可读存储器中的指令产生包括指令装置的制造品，该指令装置实现在流程图一个流程或多个流程和 / 或方框图一个方框或多个方框中指定的功能。

这些计算机程序指令也可装载到计算机或其他可编程数据处理设备上，使得在计算机
15 或其他可编程设备上执行一系列操作步骤以产生计算机实现的处理，从而在计算机或其他可编程设备上执行的指令提供用于实现在流程图一个流程或多个流程和 / 或方框图一个方框或多个方框中指定的功能的步骤。

在一个典型的配置中，计算设备包括一个或多个处理器(central processing unit, CPU)、输入/输出接口、网络接口和内存。

20 内存可能包括计算机可读介质中的非永久性存储器，随机存取存储器 (Random Access Memory, RAM)和/或非易失性内存等形式，如只读存储器(Read Only Memory, ROM)或闪存(flash RAM)。内存是计算机可读介质的示例。

计算机可读介质包括永久性和非永久性、可移动和非可移动媒体可以由任何方法或技术来实现信息存储。信息可以是计算机可读指令、数据结构、程序的模块或其他数据。计
25 算机的存储介质的例子包括，但不限于相变化内存(Phase Change RAM, PRAM)、静态随机存取存储器(Static Random-Access Memory, SRAM)、动态随机存取存储器(Dynamic Random Access Memory, DRAM)、其他类型的随机存取存储器(Random Access Memory, RAM)、只读存储器(Read Only Memory, ROM)、电可擦除可编程只读存储器(Electrically-Erasable Programmable Read-Only Memory, EEPROM)、快闪记忆体或其他内存技术、只读光盘只读
30 存储器(CD-ROM)、数字多功能光盘(Digital versatile disc, DVD)或其他光学存储、磁盒式磁带，磁带磁盘存储或其他磁性存储设备或任何其他非传输介质，可用于存储可以被计算设备访问的信息。按照本文中的界定，计算机可读介质不包括暂存电脑可读媒体(transitory

media), 如调制的数据信号和载波。

还需要说明的是, 术语“包括”、“包含”或者其任何其他变体意在涵盖非排他性的包含, 从而使得包括一系列要素的过程、方法、商品或者设备不仅包括那些要素, 而且还包括没有明确列出的其他要素, 或者是还包括为这种过程、方法、商品或者设备所固有的要素。

- 5 在没有更多限制的情况下, 由语句“包括一个……”限定的要素, 并不排除在包括所述要素的过程、方法、商品或者设备中还存在另外的相同要素。

以上所述仅为本申请的实施例而已, 并不用于限制本申请。对于本领域技术人员来说, 本申请可以有各种更改和变化。凡在本申请的精神和原理之内所作的任何修改、等同替换、改进等, 均应包含在本申请的权利要求范围之内。

权 利 要 求

- 1、一种存储卷挂载方法，其特征在于，包括：
- 确定被目标应用实例绑定且待挂载到宿主机目录树上的目标存储卷，所述宿主机目录树是所述目标应用实例所在宿主机的目录树；
- 5 若所述宿主机目录树上包含所述目标存储卷的根挂载点，针对所述目标应用实例在所述宿主机目录树中创建指向所述根挂载点的子挂载点；
- 基于所述子挂载点以子树挂载方式将所述目标存储卷挂载到所述根挂载点上，以使所述目标应用实例通过所述根挂载点访问所述目标存储卷。
- 2、根据权利要求1所述的方法，其特征在于，还包括：
- 10 若所述宿主机目录树上未包含所述目标存储卷的根挂载点，在创建所述子挂载点之前，在所述宿主机目录树中创建所述目标存储卷的根挂载点。
- 3、根据权利要求1所述的方法，其特征在于，所述目标存储卷能够被宿主机上的多个应用实例绑定，且所述多个应用实例包括所述目标应用实例。
- 4、根据权利要求1所述的方法，其特征在于，还包括：
- 15 根据所述目标存储卷的标识在所述宿主机目录树对应的挂载列表中进行匹配，所述挂载列表中记录有已挂在到所述宿主机目录树上的存储卷与根挂载点之间的映射关系；
- 若匹配到所述目标存储卷的标识，将与所述目标存储卷的标识对应的根挂载点作为所述目标存储卷的根挂载点；
- 若未匹配到所述目标存储卷的标识，确定所述宿主机目录树上未包含所述目标存储卷
- 20 的根挂载点。
- 5、根据权利要求1所述的方法，其特征在于，针对所述目标应用实例在所述宿主机目录树中创建指向所述根挂载点的子挂载点，包括：
- 从所述目标应用实例对应的配置文件中，获取所述目标应用实例在所述宿主机目录树中对应的目录位置；
- 25 将所述目录位置作为子挂载点，并建立所述子挂载点与所述根挂载点之间的指向关系。
- 6、根据权利要求5所述的方法，其特征在于，建立所述子挂载点与所述根挂载点之间的指向关系，包括：
- 将所述根挂载点作为所述子挂载点的父挂载点，并将所述根挂载点的标识记录所述子挂载点的属性信息中。
- 30 7、根据权利要求1所述的方法，其特征在于，在基于所述子挂载点以子树挂载方式将所述目标存储卷挂载到所述根挂载点上之后，还包括：
- 根据设定的次数更新步长，增加所述根挂载点的引用次数，其中，所述次数更新步长为正整数。
- 8、根据权利要求7所述的方法，其特征在于，还包括：
- 35 在所述目标应用实例从所述宿主机上销毁的情况下，根据所述次数更新步长，减少所

述根挂载点的引用次数；以及

在所述根挂载点的引用次数为预设初始值的情况下，从所述宿主机目录树上卸载所述根挂载点。

9、根据权利要求2所述的方法，其特征在于，还包括：

5 根据所述目标存储卷的根挂载点为所述目标存储卷分配缓存空间，并将所述目标存储卷的缓存空间与所述目标存储卷的根挂载点进行绑定。

10、根据权利要求9所述的方法，其特征在于，还包括：

响应所述目标应用实例对所述目标存储卷的访问操作，确定与所述目标存储卷的根挂载点绑定的所述目标存储卷的缓存空间；

10 若所述目标存储卷的缓存空间未包含所述访问操作所需的数据，根据所述目标存储卷在所述宿主机目录树上的根挂载点和子挂载点，生成所述目标存储卷的访问路径；

根据所述访问路径，对所述目标存储卷执行相应类型的访问操作。

11、一种存储卷访问方法，其特征在于，包括：

15 响应目标应用实例对目标存储卷的访问操作，根据所述目标存储卷在宿主机目录树上的根挂载点和子挂载点，生成所述目标存储卷的访问路径；

根据所述访问路径，对所述目标存储卷执行相应类型的访问操作，其中，所述目标存储卷是被所述目标应用实例绑定且挂载到所述宿主机目录树上的存储卷。

12、根据权利要求11所述的方法，其特征在于，在生成所述目标存储卷的访问路径之前，还包括：

20 确定与所述目标存储卷的根挂载点绑定的所述目标存储卷的缓存空间；

若所述目标存储卷的缓存空间未包含所述访问操作所需的数据，则执行生成所述目标存储卷的访问路径的步骤。

13、一种计算机设备，其特征在于，包括：存储器和处理器；所述存储器，用于存储计算机程序；所述处理器耦合至所述存储器，用于执行所述计算机程序以用于执行权利要求1-10以及权利要求11-12中任一项所述方法中的步骤。

25 14、一种存储有计算机程序的计算机可读存储介质，其特征在于，当所述计算机程序被处理器执行时，致使所述处理器能够实现权利要求1-10以及权利要求11-12中任一项所述方法中的步骤。