



(12) 发明专利申请

(10) 申请公布号 CN 105260231 A

(43) 申请公布日 2016. 01. 20

(21) 申请号 201510745216. 5

(22) 申请日 2015. 11. 03

(71) 申请人 国云科技股份有限公司

地址 523808 广东省东莞市松山湖科技产业
园区松科苑 14 号楼

(72) 发明人 刘勇彬 杨松 莫展鹏 季统凯

(74) 专利代理机构 广东莞信律师事务所 44332
代理人 余伦

(51) Int. Cl.

G06F 9/455(2006. 01)

G06F 3/06(2006. 01)

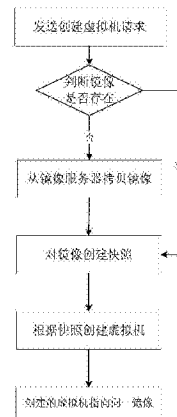
权利要求书1页 说明书3页 附图1页

(54) 发明名称

一种降低物理磁盘 I/O 读写的方法

(57) 摘要

本发明涉及云计算技术领域,特别是指一种降低物理磁盘 I/O 读写的方法。本发明包括:发送创建虚拟机请求;然后,在所在物理节点检测创建的虚拟机要求操作系统镜像是否存在,若存在,对存在镜像创建相应快照,从镜像服务器拷贝相应镜像到物理节点;最后,再根据快照创建虚拟机。本发明的方法降低了虚拟机对物理磁盘 I/O 的读写;提高存储空间利用率;可以用于解决物理磁盘 I/O 读写瓶颈的情形。



1. 一种降低物理磁盘 I/O 读写的方法,其特征在于:所述的方法包括如下步骤:
步骤 1:发送创建虚拟机请求;
步骤 2:在所在物理节点检测创建的虚拟机要求操作系统镜像是否存在,若存在,跳转到步骤 4,若不存在,执行步骤 3;
步骤 3:从镜像服务器拷贝相应镜像到物理节点;
步骤 4:对存在镜像创建相应快照;
步骤 5:根据快照创建虚拟机。
2. 根据权利要求 1 所述的降低物理磁盘 I/O 读写的方法,其特征在于:所述的发送创建虚拟机请求,是指用户从云平台管理界面发起。
3. 根据权利要求 1 所述的降低物理磁盘 I/O 读写的方法,其特征在于:所述的操作系统镜像是指操作系统安装光盘上所有数据的克隆文件,一般镜像文件后缀是 ". ISO"。
4. 根据权利要求 1 所述的降低物理磁盘 I/O 读写的方法,其特征在于:所述的从镜像服务器是指专门存放各种操作系统镜像的服务器。
5. 根据权利要求 3 所述的降低物理磁盘 I/O 读写的方法,其特征在于:所述的从镜像服务器是指专门存放各种操作系统镜像的服务器。
6. 根据权利要求 1 至 5 任一项所述的降低物理磁盘 I/O 读写的方法,其特征在于:所述的快照是指定镜像的一个完全可用拷贝,该拷贝包括相应数据在某个时间点(拷贝开始的时间点)的映像,所述的快照是指向保存在存储设备中的镜像的引用标记或指针。
7. 根据权利要求 1 至 5 任一项所述的降低物理磁盘 I/O 读写的方法,其特征在于:所述的根据快照创建虚拟机,即将虚拟机后期由于用户个人修改引起的数据变化记录到快照里面,不影响快照指向的系统镜像,从而使得个人数据与系统镜像分离。
8. 根据权利要求 6 项所述的降低物理磁盘 I/O 读写的方法,其特征在于:所述的根据快照创建虚拟机,即将虚拟机后期由于用户个人修改引起的数据变化记录到快照里面,不影响快照指向的系统镜像,从而使得个人数据与系统镜像分离。

一种降低物理磁盘 I/O 读写的方法

技术领域

[0001] 本发明涉及云计算技术领域,特别是指一种降低物理磁盘 I/O 读写的方法。

背景技术

[0002] 在云计算环境下,由于虚拟机被广泛使用,随着物理机性能的提升,一台物理机上承载的虚拟机数量越来越多。因此,物理磁盘的 I/O 逐渐成为承载数据量的瓶颈之一,目前在磁盘 I/O 上采取的降低物理磁盘 I/O 读写的方法,一般有如下:

[0003] 1、采用高性能的存储设备,提高单个物理节点磁盘 I/O 性能。

[0004] 2、限制单个物理节点上虚拟机数量。

[0005] 3、将对于 I/O 有较高要求的应用均匀放置在不同的物理节点上,避免出现 I/O 争抢的现象。

[0006] 上述方法存在以下弊端:

[0007] 1、成本较高,同样容量的存储,高性能存储设备价格是低端存储设备的 3-5 倍;

[0008] 2、限制单台物理节点虚拟机数量,会导致物理节点计算资源浪费。

[0009] 3、均匀分配高 I/O 应用,提高运维成本。

发明内容

[0010] 本发明解决的技术问题在于提供一种降低物理磁盘 I/O 读写的方法,解决传统方法存在的不足,为降低虚拟机对物理磁盘 I/O 的读写,提高存储空间利用率,解决物理磁盘 I/O 读写瓶颈提供一种可行的解决方案。

[0011] 本发明解决上述问题,包括以下步骤:

[0012] 所述的方法包括如下步骤:

[0013] 步骤 1:发送创建虚拟机请求;

[0014] 步骤 2:在所在物理节点检测创建的虚拟机要求操作系统镜像是否存在,若存在,跳转到步骤 4,若不存在,执行步骤 3;

[0015] 步骤 3:从镜像服务器拷贝相应镜像到物理节点;

[0016] 步骤 4:对存在镜像创建相应快照;

[0017] 步骤 5:根据快照创建虚拟机。

[0018] 所述的发送创建虚拟机请求,是指用户从云平台管理界面发起。

[0019] 所述的操作系统镜像是指操作系统安装光盘上所有数据的克隆文件,一般镜像文件后缀是 ".ISO"。

[0020] 所述的从镜像服务器是指专门存放各种操作系统镜像的服务器。

[0021] 所述的快照是指定镜像的一个完全可用拷贝,该拷贝包括相应数据在某个时间点(拷贝开始的时间点)的映像,所述的快照是指向保存在存储设备中的镜像的引用标记或指针。

[0022] 所述的根据快照创建虚拟机,即将虚拟机后期由于用户个人修改引起的数据变化

记录到快照里面,不影响快照指向的系统镜像,从而使得个人数据与系统镜像分离。

[0023] 本发明的方法能降低虚拟机对物理磁盘 IO 的读写,相同操作系统的虚拟机公用同一系统镜像,当虚拟机并发启动时,虚拟机读取同一系统镜像,从而降低对物理磁盘 IO 的读写;降低镜像存储空间,相同操作系统的虚拟机公用同一系统镜像,从而节省每台虚拟机对应独立系统镜像的存储空间。

附图说明

[0024] 下面结合附图对本发明进一步说明:

[0025] 图 1 为本发明的流程图。

具体实施方式

[0026] 本发明的实施方式有多种,这里将基于 KVM 虚拟化实现方式为例说明其中一种实现方法,流程图如图 1 所示,具体实施过程如下

[0027] 1、发送创建虚拟机请求;

[0028]

```
/**
 * createInstance 创建虚拟机.
 * @param vm 虚拟机 bean, 包含操作系统类型、CPU 大小、内存大小、硬盘大小、虚拟机 ID、
虚拟机名称
 * @param success 是否成功, 1 表示成功, 0 表示失败
 */
public int createInstance(VmInstance vm) {
    int success= gComputerService. createInstance (vm);
    return success;
}
```

[0029] 2、在所在物理节点检测创建的虚拟机要求操作系统镜像是否存在,若存在,顺序执行 4,若不存在,跳转到步骤 3;

[0030]

```
/**
 * isImageExist;检测镜像是否存在.
 * @param imageId 操作系统镜像 ID
 * @param success 是否成功, true 表示存在, false 表示不存在
 */
public boolean isImageExist (int imageId) {
    boolean success= gComputerService. isImageExist (imageId);
    return success;
}
```

[0031] 3、从镜像服务器拷贝相应镜像到物理节点;

[0032]

```
/**
 * scpImageToNode;拷贝镜像到指定物理机节点.
```

[0033]

```

    * @param desIp 镜像拷贝目标物理节点
    * @param imageId 操作系统镜像 ID
    * @param success 是否成功, true 表示成功, false 表示失败
    */
    public boolean isImageExist (int imageId, String desIp) {
        boolean success= gComputerService.isImageExist (imageId, desIp);
        return success;
    }
}

```

[0034] 4、对存在镜像创建相应快照；

[0035]

```

    /**
    * snapShot;根据镜像创建快照.
    * @param imageSrc 镜像路径
    * @param snapName 快照名称
    * @param success 是否成功, true 表示成功, false 表示失败
    */
    public boolean snapShot(String snapName, String imageSrc) {
        boolean success= gComputerService. snapShot (snapName ,imageSrc);
        return success;
    }
}

```

[0036] 底层命令 #qemu-img snapshot-c\$1\$2

[0037] \$1 :快照名称

[0038] \$2 :镜像路径

[0039] 5、再根据快照创建虚拟机

[0040] 在 libvirt.xml 文件里面设定镜像快照路径：

[0041]

```

    <disk device="disk" type="file">
        <driver name="qemu" cache="none" type="qcow2"/>
        <source file="指向快照路径"/>
        <target bus="virtio" dev="vda"/>
    </disk>

```

[0042] 通过底层命令创建虚拟机 # :virsh create libvirt。

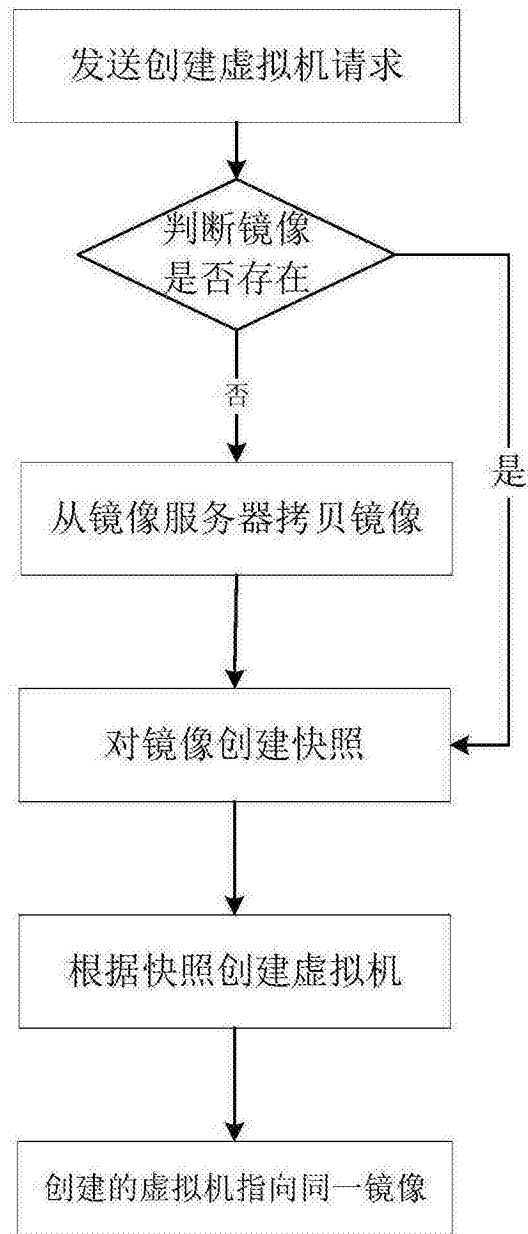


图 1