



(12)发明专利申请

(10)申请公布号 CN 111352647 A

(43)申请公布日 2020.06.30

(21)申请号 202010119885.2

(22)申请日 2020.02.26

(71)申请人 平安科技(深圳)有限公司

地址 518033 广东省深圳市福田区福田街
道福安社区益田路5033号平安金融中
心23楼

(72)发明人 瞿盛辉

(74)专利代理机构 北京市京大律师事务所

11321

代理人 刘挽澜

(51)Int.Cl.

G06F 8/656(2018.01)

G06F 9/455(2006.01)

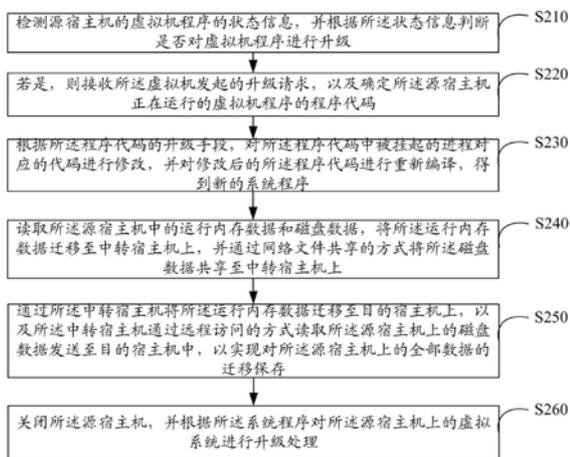
权利要求书2页 说明书11页 附图7页

(54)发明名称

虚拟机升级方法、装置、设备及存储介质

(57)摘要

本发明涉及云技术领域,公开了一种虚拟机升级方法,通过根据源宿主机本身的虚拟系统的程序代码进行修改,得到新的系统程序,并基于修改后的系统程序对源宿主机进行升级,并且在升级过程中,将源宿主机中的内存和磁盘两部分的数据分开通过中转宿主机来实现迁移;本发明还提供了一种虚拟机升级装置、设备及计算机可读存储介质,这样的实现方式可以不用关闭虚拟机直接升级虚拟系统,同时热迁移虚拟机到其他宿主机上,避免了升级过程中可能存在的挂起问题。



1. 一种虚拟机升级方法,应用于服务器,其特征在于,所述虚拟机升级方法包括以下步骤:

检测源宿主机的虚拟机程序的状态信息,并根据所述状态信息判断是否对虚拟机程序进行升级;

若是,则接收所述虚拟机发起的升级请求,以及确定所述源宿主机正在运行的虚拟机程序的程序代码,其中,所述升级请求包括对所述程序代码的升级手段;

根据所述程序代码的升级手段,对所述程序代码中被挂起的进程对应的代码进行修改,并对修改后的所述程序代码进行重新编译,得到新的系统程序;

读取所述源宿主机中的运行内存数据和磁盘数据,将所述运行内存数据迁移至中转宿主机上,并通过网络文件共享的方式将所述磁盘数据共享至中转宿主机上;

通过所述中转宿主机将所述运行内存数据迁移至目的宿主机上,以及所述中转宿主机通过远程访问的方式读取所述源宿主机的磁盘数据发送至目的宿主机中,以实现所述源宿主机的全部数据的迁移保存;

关闭所述源宿主机,并根据所述系统程序对所述源宿主机的虚拟系统进行升级处理。

2. 如权利要求1所述的虚拟机升级方法,其特征在于,所述状态信息为所述虚拟机程序当前运行的进程状态,所述根据所述状态信息判断是否需要所述虚拟机程序进行升级的步骤包括:

判断所述虚拟机程序中的进程是否出现了挂起状态,其中,所述挂起状态为所述进程在同一程序节点上停留的时间不小于N倍该程序节点的预设执行时间,N为大于3的整数;

若判断为挂起状态,则确定所述虚拟机程序需要进行升级。

3. 如权利要求2所述的虚拟机升级方法,其特征在于,所述根据所述程序代码的升级手段,对所述程序代码中被挂起的进程对应的代码进行修改,并对修改后的所述程序代码进行重新编译,得到新的系统程序的步骤包括:

检测出现挂机状态对应的程序节点在所述程序代码中位置,并根据所述位置获取所述程序节点所在的进程代码段;

判断所述进程代码段是否属于所述程序代码的主进程程序;

根据判断的结果和所述程序代码的升级手段选择对应的代码修改策略;

根据所述代码修改策略对所述进程代码段进行升级,生成新的系统程序。

4. 如权利要求3所述的虚拟机升级方法,其特征在于,所述根据判断的结果和所述程序代码的升级手段确定对应的代码修改策略包括:

若判断所述进程代码段属于主进程程序时,则选择的所述代码修改策略为增加跳转函数;

若判断所述进程代码段不属于主进程程序时,则选择的所述代码修改策略为删除代码段。

5. 如权利要求4所述的虚拟机升级方法,其特征在于,所述根据所述代码修改策略对所述进程代码段进行升级,生成新的系统程序包括:

若所述代码修改策略为删除代码段,则将所述进程代码段从所述程序代码中删除,并重新编译生成所述系统程序;

若所述代码修改策略为增加跳转函数,则在所述进程代码段中出现挂起状态的程序节点之后增加时间判断函数,通过所述时间判断函数对所述进程代码的挂机状态的停留时间进行判断,并重新编译生成所述系统程序。

6. 如权利要求5所述的虚拟机升级方法,其特征在于,若所述代码修改策略为增加跳转函数时,所述根据所述代码修改策略对所述进程代码段进行升级,生成新的系统程序的步骤,还包括:

在出现挂机状态的所述进程代码段之前的一个程序节点上设置程序跳转指针,并重新编译生成所述系统程序,其中所述程序跳转指针用于在所述时间判断函数确定所述停留时间超出预设执行时长时,将所述虚拟机从当前的系统程序跳转至其他的任务程序执行。

7. 如权利要求1-6任一项所述的虚拟机升级方法,其特征在于,在根据所述程序代码的升级手段,对所述程序代码中被挂起的进程对应的代码进行修改,并对修改后的所述程序代码进行重新编译,得到新的系统程序的步骤之后,还包括:

根据所述系统程序分别对所述中转宿主机和目的宿主机中的虚拟机程序进行升级;

当所述中转宿主机和目的宿主机升级完成后,执行将所述源宿主机上的所有数据通过所述中转宿主机迁移至所述目的宿主机上的步骤。

8. 一种虚拟机升级装置,其特征在于,所述虚拟机升级装置包括:

判断模块,用于检测源宿主机的虚拟机程序的状态信息,并根据所述状态信息判断是否对虚拟机程序进行升级;

读取模块,用于在判断虚拟机程序需要升级时,接收所述虚拟机发起的升级请求,以及确定所述源宿主机正在运行的虚拟机程序的程序代码,其中,所述升级请求包括对所述程序代码的升级手段;

编译模块,用于根据所述程序代码的升级手段,对所述程序代码中被挂起的进程对应的代码进行修改,并对修改后的所述程序代码进行重新编译,得到新的系统程序;

数据共享模块,用于读取所述源宿主机中的运行内存数据和磁盘数据,将所述运行内存数据迁移至中转宿主机上,并通过网络文件共享的方式将所述磁盘数据共享至中转宿主机上;

数据转移模块,用于通过所述中转宿主机将所述运行内存数据迁移至目的宿主机上,以及所述中转宿主机通过远程访问的方式读取所述源宿主机上的磁盘数据发送至目的宿主机中,以实现所述源宿主机上的全部数据的迁移保存;

升级模块,用于关闭所述源宿主机,并根据所述系统程序对所述源宿主机上的虚拟系统进行升级处理。

9. 一种虚拟机升级设备,其特征在于,所述虚拟机升级设备包括:存储器、处理器以及存储在所述存储器上并可在所述处理器上运行的虚拟机升级程序,所述虚拟机升级程序被所述处理器执行时实现如权利要求1-7中任一项所述的虚拟机升级方法的步骤。

10. 一种计算机可读存储介质,其特征在于,所述计算机可读存储介质上存储有虚拟机升级程序,所述虚拟机升级程序被所述处理器执行时实现如权利要求1-7中任一项所述的虚拟机升级方法的步骤。

虚拟机升级方法、装置、设备及存储介质

技术领域

[0001] 本发明涉及云技术领域,尤其涉及一种虚拟机升级方法、装置、设备及计算机可读存储介质。

背景技术

[0002] 在云计算环境中,虚拟模拟器qemu、虚拟化模块kvm和qcow2格式盘的架构是当前实现虚拟系统所应用较为广泛的实现方式之一,在使用该架构实现虚拟化过程中,热迁移是qemu提供的一个强大功能,但是在当前热迁移实现数据迁移的过程中,会通过构建bitmap来标示是否需要迁移数据,而在构建bitmap时,qemu会先通过调用format driver (qcow2 driver)获取本次所要探测数据区域数据分配实际情况;当发现存在数据分配,会再次调用相应的file driver (raw driver)来探测在文件对应的实际区域是否真实存在数据。以跳过文件中实际并没有写的区域,以减少迁移时数据传输。在file driver探测是否存在空洞时,通过调用lseek系统调用来判断文件是否存在数据空洞。

[0003] 在查找空洞时会从传入的开始位置一直查找到空洞或查找到文件结束为止,而进入lseek的时间长短与文件(也就是磁盘)大小和文件碎片多少成正比,也就是文件越大碎片越多在lseek中的时间就越长。这就导致当虚拟机单个磁盘很大同时磁盘碎片很多时当前热迁移过程就长时间呆在lseek系统调用中,导致qemu主进程无法去响应其他的事务,对应虚拟机来说体现的就是虚拟机主流程被挂住,即是可以理解为是虚拟化实现的主流程进入了深度循环,这就导致了系统无法在调度资源来实现其他的业务了,这就大大降低了虚拟系统对于业务的处理能力;而在目前的解决方式中,只能通过重启虚拟机或者是关闭虚拟机进行qemu升级的方式来实现,而不管是重启还是关闭虚拟机升级,都会造成数据的丢失,拖延后续系统的业务处理,影响用户的使用。

发明内容

[0004] 本发明的主要目的在于提供一种虚拟机升级方法、装置、设备及计算机可读存储介质,旨在解决现有的虚拟机在升级过程中会被挂住的技术问题。

[0005] 为实现上述目的,本发明提供一种虚拟机升级方法,应用于服务器,所述虚拟机升级方法包括以下步骤:

[0006] 检测源宿主机的虚拟机程序的状态信息,并根据所述状态信息判断是否对虚拟机程序进行升级;

[0007] 若是,则接收所述虚拟机发起的升级请求,以及确定所述源宿主机正在运行的虚拟机程序的程序代码,其中,所述升级请求包括对所述程序代码的升级手段;

[0008] 根据所述程序代码的升级手段,对所述程序代码中被挂起的进程对应的代码进行修改,并对修改后的所述程序代码进行重新编译,得到新的系统程序;

[0009] 读取所述源宿主机中的运行内存数据和磁盘数据,将所述运行内存数据迁移至中转宿主机上,并通过网络文件共享的方式将所述磁盘数据共享至中转宿主机上;

[0010] 通过所述中转宿主机将所述运行内存数据迁移至目的宿主机上,以及所述中转宿主机通过远程访问的方式读取所述源宿主机上的磁盘数据发送至目的宿主机中,以实现所述源宿主机上的全部数据的迁移保存;

[0011] 关闭所述源宿主机,并根据所述系统程序对所述源宿主机上的虚拟系统进行升级处理。

[0012] 可选地,所述状态信息为所述虚拟机程序当前运行的进程状态;

[0013] 所述根据所述状态信息判断是否需要所述虚拟机程序进行升级的步骤包括:

[0014] 判断所述虚拟机程序中的进程是否出现了挂起状态,其中,所述挂起状态为所述进程在同一程序节点上停留的时间不小于N倍该程序节点的预设执行时间,N为大于3的整数;

[0015] 若判断为挂起状态,则确定所述虚拟机程序需要进行升级。

[0016] 可选地,所述根据所述程序代码的升级手段,对所述程序代码中被挂起的进程对应的代码进行修改,并对修改后的所述程序代码进行重新编译,得到新的系统程序的步骤包括:

[0017] 检测出现挂机状态对应的程序节点在所述程序代码中位置,并根据所述位置获取所述程序节点所在的进程代码段;

[0018] 判断所述进程代码段是否属于所述程序代码的主进程程序;

[0019] 根据判断的结果和所述程序代码的升级手段选择对应的代码修改策略;

[0020] 根据所述代码修改策略对所述进程代码段进行升级,生成新的系统程序。

[0021] 可选地,所述根据判断的结果和所述程序代码的升级手段确定对应的代码修改策略包括:

[0022] 若判断所述进程代码段属于主进程程序时,则选择的所述代码修改策略为增加跳转函数;

[0023] 若判断所述进程代码段不属于主进程程序时,则选择的所述代码修改策略为删除代码段。

[0024] 可选地,所述根据所述代码修改策略对所述进程代码段进行升级,生成新的系统程序包括:

[0025] 若所述代码修改策略为删除代码段,则将所述进程代码段从所述程序代码中删除,并重新编译生成所述系统程序;

[0026] 若所述代码修改策略为增加跳转函数,则在所述进程代码段中出现挂起状态的程序节点之后增加时间判断函数,通过所述时间判断函数对所述进程代码的挂机状态的停留时间进行判断,并重新编译生成所述系统程序。

[0027] 可选地,若所述代码修改策略为增加跳转函数时,所述根据所述代码修改策略对所述进程代码段进行升级,生成新的系统程序的步骤,还包括:

[0028] 在出现挂机状态的所述进程代码段之前的一个程序节点上设置程序跳转指针,并重新编译生成所述系统程序,其中所述程序跳转指针用于在所述时间判断函数确定所述停留时间超出预设执行时长时,将所述虚拟机从当前的系统程序跳转至其他的任务程序执行。

[0029] 可选地,在根据所述程序代码的升级手段,对所述程序代码中被挂起的进程对应

的代码进行修改,并对修改后的所述程序代码进行重新编译,得到新的系统程序的步骤之后,还包括:

[0030] 根据所述系统程序分别对所述中转宿主机和目的宿主机中的虚拟机程序进行升级;

[0031] 当所述中转宿主机和目的宿主机升级完成后,执行将所述源宿主机上的所有数据通过所述中转宿主机迁移至所述目的宿主机上的步骤。

[0032] 此外,为实现上述目的,本发明还提供一种虚拟机升级装置,所述虚拟机升级装置包括:

[0033] 判断模块,用于检测源宿主机的虚拟机程序的状态信息,并根据所述状态信息判断是否对虚拟机程序进行升级;

[0034] 读取模块,用于在判断虚拟机程序需要升级时,接收所述虚拟机发起的升级请求,以及确定所述源宿主机正在运行的虚拟机程序的程序代码,其中,所述升级请求包括对所述程序代码的升级手段;

[0035] 编译模块,用于根据所述程序代码的升级手段,对所述程序代码中被挂起的进程对应的代码进行修改,并对修改后的所述程序代码进行重新编译,得到新的系统程序;

[0036] 数据共享模块,用于读取所述源宿主机中的运行内存数据和磁盘数据,将所述运行内存数据迁移至中转宿主机上,并通过网络文件共享的方式将所述磁盘数据共享至中转宿主机上;

[0037] 数据转移模块,用于通过所述中转宿主机将所述运行内存数据迁移至目的宿主机上,以及所述中转宿主机通过远程访问的方式读取所述源宿主机上的磁盘数据发送至目的宿主机中,以实现所述源宿主机上的全部数据的迁移保存;

[0038] 升级模块,用于关闭所述源宿主机,并根据所述系统程序对所述源宿主机上的虚拟系统进行升级处理。

[0039] 可选地,所述状态信息为所述虚拟机程序当前运行的进程状态;

[0040] 所述判断模块用于判断所述虚拟机程序中的进程是否出现了挂起状态,其中,所述挂起状态为所述进程在同一程序节点上停留的时间不小于N倍该程序节点的预设执行时间,N为大于3的整数;若判断为挂起状态,则确定所述虚拟机程序需要进行升级。

[0041] 可选地,所述编译模块包括代码检测单元、修改单元和编译单元,其中:

[0042] 所述代码检测单元用于检测出现挂机状态对应的程序节点在所述程序代码中位置,并根据所述位置获取所述程序节点所在的进程代码段;判断所述进程代码段是否属于所述程序代码的主进程程序;

[0043] 所述修改单元用于根据判断的结果和所述程序代码的升级手段选择对应的代码修改策略;

[0044] 所述编译单元用于根据所述代码修改策略对所述进程代码段进行升级,生成新的系统程序。

[0045] 可选地,所述修改单元用于,在判断所述进程代码段属于主进程程序时,选择的所述代码修改策略为增加跳转函数;或者用于在判断所述进程代码段不属于主进程程序时,选择的所述代码修改策略为删除代码段。

[0046] 可选地,所述编译单元用于在所述代码修改策略为删除代码段时,将所述进程代

码段从所述程序代码中删除,并重新编译生成所述系统程序;或者用于在所述代码修改策略为增加跳转函数时,在所述进程代码段中出现挂起状态的程序节点之后增加时间判断函数,通过所述时间判断函数对所述进程代码的挂机状态的停留时间进行判断,并重新编译生成所述系统程序。

[0047] 可选地,若所述代码修改策略为增加跳转函数时,所述修改单元还用于在出现挂机状态的所述进程代码段之前的一个程序节点上设置程序跳转指针,并重新编译生成所述系统程序,其中所述程序跳转指针用于在所述时间判断函数确定所述停留时间超出预设执行时长时,将所述虚拟机从当前的系统程序跳转至其他的任务程序执行。

[0048] 可选地,所述升级模块还用于根据所述系统程序分别对所述中转宿主机和目的宿主机中的虚拟机程序进行升级;

[0049] 所述数据迁移模块还用于在所述中转宿主机和目的宿主机升级完成后,执行将所述源宿主机的所有数据通过所述中转主机迁移至所述目的宿主机上的步骤。

[0050] 此外,为实现上述目的,本发明还提供了一种虚拟机升级设备,所述虚拟机升级设备包括:存储器、处理器以及存储在所述存储器上并可在所述处理器上运行的虚拟机升级程序,所述虚拟机升级程序被所述处理器执行时实现如上任一项所述的虚拟机升级方法的步骤。

[0051] 此外,为实现上述目的,本发明还提供了一种计算机可读存储介质,所述计算机可读存储介质上存储有虚拟机升级程序,所述虚拟机升级程序被所述处理器执行时实现如上任一项所述的虚拟机升级方法的步骤。

[0052] 本发明通过的方法,通过根据源宿主机本身的虚拟系统的程序代码进行修改,得到新的系统程序,并基于修改后的系统程序对源宿主机进行升级,并且在升级过程中,将源宿主机中的内存和磁盘两部分的数据分开通过中转主机来实现迁移,这样的实现方式可以不用关闭虚拟机直接升级虚拟系统,同时热迁移虚拟机到其他宿主机上,避免了升级过程中可能存在的hang住问题。

附图说明

[0053] 图1为本发明实施例方案涉及的服务器的运行环境的结构示意图;

[0054] 图2为本发明提供的虚拟机升级方法第一实施例的流程示意图;

[0055] 图3为本发明提供的虚拟机升级方法第二实施例的流程示意图;

[0056] 图4为本发明提供的虚拟机升级方法第三实施例的流程示意图;

[0057] 图5为本发明提供的对源宿主机升级时的数据转移示意图;

[0058] 图6为本发明提供的进程代码修改示意图;

[0059] 图7为本发明提供的进程代码修改示意图;

[0060] 图8为本发明虚拟机升级装置的功能模块示意图。

[0061] 本发明目的的实现、功能特点及优点将结合实施例,参照附图做进一步说明。

具体实施方式

[0062] 应当理解,此处所描述的具体实施例仅用以解释本发明,并不用于限定本发明。

[0063] 本发明提供一种数据同步设备。

[0064] 参照图1,图1为本发明实施例方案涉及的服务器的运行环境的结构示意图。

[0065] 如图1所示,该服务器包括:处理器101,例如CPU,通信总线102、用户接口103,网络接口104,存储器105。其中,通信总线102用于实现这些组件之间的连接通信。用户接口103可以包括显示屏(Display)、输入单元比如键盘(Keyboard),网络接口104可选地可以包括标准的有线接口、无线接口(如WI-FI接口)。存储器105可以是高速RAM存储器,也可以是稳定的存储器(non-volatile memory),例如磁盘存储器。存储器105可选地还可以是独立于前述处理器101的存储装置。

[0066] 本领域技术人员可以理解,图1中示出的服务器的硬件结构也应用于虚拟机上,其并不构成对本发明中的虚拟机、虚拟机升级装置和设备的限定,可以包括比图示更多或更少的部件,或者组合某些部件,或者不同的部件布置。

[0067] 如图1所示,作为一种计算机可读存储介质的存储器105中可以包括操作系统、网络通信程序模块、用户接口程序模块以及用于实现虚拟机升级程序。其中,操作系统是管理和控制虚拟机(源宿主机、中转宿主机和目的宿主机)中各模块的通信以及数据的获取处理,以及控制存储器中实现软件资源调用的程序,支持虚拟机升级程序以及其它软件和/或程序的运行。

[0068] 在图1所示的服务器的硬件结构中,网络接口104主要用于接入网络;用户接口103主要用于是否虚拟机中的系统程序流程是否出现挂机的状况,而处理器101可以用于调用存储器105中存储的虚拟机升级程序,并执行以下虚拟机升级方法的各实施例的操作。

[0069] 基于上述业务系统的硬件结构,提出本发明规则调用架构系统的配置方法的各个实施例。

[0070] 参照图2,图2为本发明实施例提供的虚拟机升级方法的流程图。在本实施例中,该方法是通过判断虚拟机是否会出现数据迁移的挂机,若出现,则通过修改数据迁移的程序流程的方式来对虚拟机的系统程序进行升级,从而解决挂机的的问题,该虚拟机升级方法具体包括以下步骤:

[0071] 步骤S210,检测源宿主机的虚拟机程序的状态信息,并根据所述状态信息判断是否对虚拟机程序进行升级;

[0072] 在该步骤中,这里的状态信息具体指的是源宿主机的虚拟机在执行进程过程中,由于处理数据量较大的时候,会出现各种状态,比如,进程执行过慢而导致虚拟机被hang住,无法跳出来执行其他的进程,或者是长时间停留在同一个进程中。

[0073] 而这里的源宿主机的状态信息,就是检测虚拟机在当前执行的进程中是否出现这种情况,若存在,则执行步骤S220,反之,继续监控进程状态。

[0074] 在本实施例中,所述状态信息还包括版本信息,在判断是否需要升级时,具体时间通过版本信息来确定虚拟机程序运行的总时长,根据总时长来判断是否需要升级;具体是通过版本信息中的版本号以及版本号的命名规则来计算当前的虚拟机程序运行的总时长,例如,版本号的命名规则中包含有时间和版本次数,而在实际应用中,会对虚拟机程序设置有规定的更新时间,如每个月都需要进行一次更新或者升级,而根据版本号中的时间结合当前时间来计算出虚拟机程序的运行总时长,将总时长与规定的更新或者升级时间进行比较,若大于或者等于,则判定为需要进行升级,反之,则继续监控。

[0075] 在本实施例中,所述状态信息为所述虚拟机程序当前运行的进程状态,这时,该步

骤可以通过判断对应的进程状态来实现,具体的:判断所述虚拟机程序中的进程是否出现了挂起状态,其中,所述挂起状态为所述进程在同一程序节点上停留的时间不小于N倍该程序节点的预设执行时间,N为大于3的整数;若判断为挂起状态,则确定所述虚拟机程序需要进行升级;反之,则不需要升级,返回步骤S210继续监控其他进程的状态,或者是结束处理流程。

[0076] 在实际应用中,对于程序代码的执行时间一般都是固定设置的,而每条代码执行处理完数据后,会根据数据量的大小进行计算,得到一个大概的时间长度。基于该时间长度来判断程序节点的实际执行时间,从而实现确定是都需要进行升级处理,以达到优化虚拟机系统的作用。

[0077] 步骤S220,若是,则接收所述虚拟机发起的升级请求,以及确定所述源宿主机正在运行的虚拟机程序的程序代码,其中,所述升级请求包括对所述程序代码的升级手段;

[0078] 在本实施例中,在检测到进程出现了被hang住时,根据具体的进程生成升级请求,这里的升级请求主要是用于控制服务器生成虚拟系统的升级包,具体是通过升级请求中要求的升级手段对虚拟系统的程序代码进行修改来得到。

[0079] 在判断需要升级程序代码时,还需要根据具体的状态信息来生成升级请求,其中,对于状态信息是程序代码中的主进程出现被挂起状态时,则选择删除进程代码段和在被挂起的进程节点的位置上增加一个跳转函数来将进程跳出挂起状态,执行其他的进程。

[0080] 步骤S230,根据所述程序代码的升级手段,对所述程序代码中被挂起的进程对应的代码进行修改,并对修改后的所述程序代码进行重新编译,得到新的系统程序;

[0081] 在本实施例中,对于修改程序代码具体是对程序代码中被挂起的进程对应的代码进行修改,例如将对应的代码段从程序代码中删除或者是屏蔽处理,在实际应用中,可以通过设置设置条件判断的屏蔽函数来实现,例如当检测到挂起状态的停留时间已经超过了5倍的进程执行时间,这时,则启动屏蔽函数对该进程的所有代码进行屏蔽执行,直接跳过该进程的执行,优选的跳转至执行其他的进程。

[0082] 进一步的,在修改完成后,对程序代码进行重新编译,得到新的系统程序,该系统程序即是根据上述的升级手段修改后得到的完整的程序代码。

[0083] 步骤S240,读取所述源宿主机中的运行内存数据和磁盘数据,将所述运行内存数据迁移至中转宿主机上,并通过网络文件共享的方式将所述磁盘数据共享至中转宿主机上;

[0084] 在本实施例中,源宿主机可以通过执行数据分解程序来将内存和磁盘两部分的数据进行区分开来,单独提取出内存数据和磁盘数据,而首先通过NFS将磁盘数据挂载到中转宿主机上,即是理解为是设置数据共享的方式来实现,然后在将内存的部分直接迁移到中转宿主机上,具体是迁移至中转宿主机的虚拟机上。

[0085] 在该步骤中,在迁移了内存数据之后,源宿主机上的虚拟机会被停止运行,但是源宿主机本身不会关机,仍然可以进行通过TCP/IP协议进行远程访问,具体的该远程访问仅是对磁盘数据的访问读取。

[0086] 步骤S250,通过所述中转宿主机将所述运行内存数据迁移至目的宿主机上,以及所述中转宿主机通过远程访问的方式读取所述源宿主机的磁盘数据发送至目的宿主机中,以实现所述源宿主机的全部数据的迁移保存;

[0087] 在本实施例中,中转宿主机在接收到迁移的内存数据后,根据接收到的迁移请求,将内存数据再转迁到目的宿主主机上,将内存数据在目的宿主主机上运行,而在内存数据运行的同时,请求将源宿主主机上的磁盘数据迁移或者备份至目的宿主主机的虚拟机上,而对于磁盘数据的迁移,具体是由中转宿主机通过TCP/IP访问源宿主主机的磁盘,读取磁盘中的数据文件传输给目的宿主主机上的磁盘中,从而实现对磁盘数据的迁移。

[0088] 步骤S260,关闭所述源宿主机,并根据所述系统程序对所述源宿主主机上的虚拟系统进行升级处理。

[0089] 在本实施例中,对于在对程序代码的修改过程中,还需要考虑被挂起的程序代码是否属于主进程部分,根据进程在虚拟机程序中的重要程度来选择对应的代码修改方式生成满足条件的系统程序,具体的:对于步骤S130,可以通过以下方式实现:

[0090] 检测出现挂机状态对应的程序节点在所述程序代码中位置,并根据所述位置获取所述程序节点所在的进程代码段;

[0091] 判断所述进程代码段是否属于所述程序代码的主进程程序;

[0092] 根据判断的结果和所述程序代码的升级手段选择对应的代码修改策略;

[0093] 根据所述代码修改策略对所述进程代码段进行升级,生成新的系统程序。

[0094] 在本实施例中,该虚拟机升级方法主要是正对虚拟机在进行数据迁移的处理进程的升级处理,在实际应用中,数据迁移的过程可以分为两大过程,一是对迁移数据的标注,二是对整体数据的探测确定迁移的数据并进行迁移处理,而常常出现挂起状态的,则是第二个过程,对此,本实施例中,可以通过检测执行的探测程序来确定是否属于主进程出现挂起状态;

[0095] 具体的,首先在检测到虚拟机存在挂起的状态后,通过检测虚拟机在当前所调用的驱动程序是否为文件格式驱动程序,若是,则说明当前执行的是主进程的代码流程,并且确定被挂起节点在主进程中的位置,根据节点的代码函数来选择对应的修改方式,基于修改方式对主进程进行修改即可。

[0096] 在本实施例中,所述根据判断的结果和所述程序代码的升级手段确定对应的代码修改策略包括:

[0097] 若判断所述进程代码段属于主进程程序时,则选择的所述代码修改策略为增加跳转函数;

[0098] 若判断所述进程代码段不属于主进程程序时,则选择的所述代码修改策略为删除代码段。

[0099] 在实际应用中,对于主进程的修改策略还可以是直接删除的方式进行处理。

[0100] 在本实施例中,所述根据所述代码修改策略对所述进程代码段进行升级,生成新的系统程序包括:

[0101] 若所述代码修改策略为删除代码段,则将所述进程代码段从所述程序代码中删除,并重新编译生成所述系统程序;

[0102] 若所述代码修改策略为增加跳转函数,则在所述进程代码段中出现挂起状态的程序节点之后增加时间判断函数,通过所述时间判断函数对所述进程代码的挂机状态的停留时间进行判断,并重新编译生成所述系统程序。

[0103] 在实际应用中,对于增加跳转函数的方式修改程序代码,除了采用时间判断函数

之外,还可以是其他的函数,只要能实现对进程的处理流程优化的函数,避免进程执行出现挂起状态的都可以,例如使用跳转剪头也可以,甚至还可以是采用硬件的控制也可以,当然使用硬件的前提下是,主机中存在定时器的硬件基础,通过将定时器设置在相关的进程代码中,又定时器来判断出现挂起状态的节点的执行操作,从而解决了节点停留时间过长的的问题。

[0104] 在本实施例中,若所述代码修改策略为增加跳转函数时,所述根据所述代码修改策略对所述进程代码段进行升级,生成新的系统程序的步骤,还包括:

[0105] 在出现挂机状态的所述进程代码段之前的一个程序节点上设置程序跳转指针,并重新编译生成所述系统程序,其中所述程序跳转指针用于在所述时间判断函数确定所述停留时间超出预设执行时长时,将所述虚拟机从当前的系统程序跳转至其他的任务程序执行。

[0106] 进一步的,为了优化数据迁移过程中也会出现挂起的问题,本实施例中,还包括对中转宿主机和目的宿主机的虚拟机系统进行升级,而升级采用的正是上述方式修改的系统程序,具体的:在根据所述程序代码的升级手段,对所述程序代码中被挂起的进程对应的代码进行修改,并对修改后的所述程序代码进行重新编译,得到新的系统程序的步骤之后,还包括:

[0107] 根据所述系统程序分别对所述中转宿主机和目的宿主机中的虚拟机程序进行升级;

[0108] 当所述中转宿主机和目的宿主机升级完成后,执行将所述源宿主机上的所有数据通过所述中转宿主机迁移至所述目的宿主机上的步骤。

[0109] 下面以具体的应用场景对上述的方案进行说明,以三个服务器为例说明,服务器A、服务器B和服务器C,三个服务器分别对应的是源宿主机、中转宿主机和目的宿主机,如图5所示。

[0110] 在本实施例中,对升级虚拟机的系统程序具体包括两种方式:

[0111] 第一种,通过直接删除进程的方式,得到新的系统程序;

[0112] 在实际应用中,这里的删除并不是现传统意义上的删除,而是通过设置直转的跳转指令,将对应的进程屏蔽执行,如图3所示,其具体的执行流程包括:

[0113] 步骤S310、qemu在首轮迁移本地盘(qcow2)时会构建bitmap来标示是否需要迁移数据。

[0114] 步骤S320、在构建bitmap过程中,qemu先调用相应format driver(qcow2 driver)获取本次所要探测数据区域数据分配实际情况。

[0115] 步骤S330、当发现存在数据分配,会再次调用相应的filedriver(raw driver)来探测在文件对应的实际区域是否真实存在数据。以跳过文件中实际并没有写的区域,以减少迁移时数据传输。

[0116] 步骤S340、filedriver探测是否存在空洞时,通过调用lseek系统调用来判断文件是否存在数据空洞。

[0117] 步骤S350,判断调用lseek系统调用来判断文件是否存在数据空洞时,其执行时间的长度。

[0118] 步骤S360,若大于预设值,则将获取对应的代码段,直接将其从qemu中删除。

[0119] 在该步骤中,直接删除的方式,在原来流程中直接将filedriver的处理跳过,如图6所示,将虚线框中的流程直接跳过,流程直接如标注1的箭头所示,判断完qcow2driver后就直接设置bitmap。

[0120] 当然,这里的删除也可以是可恢复的删除,例如使用代码屏蔽函数将主进程中的代码段进行屏蔽,并增加一个跳转剪头来跳过主进程代码的执行。

[0121] 步骤S370,启动数据迁移,将源宿主机上的数据迁移至目的宿主机上,并根据删除后的程序代码对源宿主机的qemu进行升级。

[0122] 在本实施例提出的虚拟机升级方法具体是针对上述流程中的步骤S320和S330,即是通过本发明提供的方法对服务器A中的虚拟机的数据迁移过程中对数据迁移分配检测的过程进行检测,当检测步骤S320和S330的执行状态在挂起状态,即是流程停留的时间过长,影响了后续的进程或者是停留的时间过长未能及时从步骤S320和S330中跳转出来执行其他的进行,而导致虚拟机的迁移出现滞留。

[0123] 第二种,增加跳转函数的方式,即是在原来流程调用rawdriever迁移,增加一个时间判断,当时间超过一定时间时就切换出去执行其他事务,执行完其他事务后再切换回来继续运行,如图7中虚线框中列出的流程,而对于增加了跳转函数后的qemu的执行流程,如图3所示,其具体流程包括:

[0124] 首先执行的是步骤S410-S450,其与步骤S310-S340相同;

[0125] 步骤S450,启动时间判断函数,计算lseek系统的执行时间的长度。

[0126] 步骤S460,若执行时间长度达到预设值,则标记进程的位置,并启动虚拟机的升级操作。

[0127] 步骤S470,启动数据迁移,将源宿主机上的数据迁移至目的宿主机上,并根据增加时间判断函数后的程序代码对源宿主机的qemu进行升级。

[0128] 在本实施例中,对于修改qemu升级的方式,具体实现步骤如下:

[0129] 1、在目标宿主机服务器C和中转服务器B上先升级去掉filedriver探测的qemu;

[0130] 2、通过NFS将源宿主服务器A上的虚拟机磁盘挂载到中转服务器B,利用NFS迁移,将虚拟机从源宿主服务器A迁移到中转服务器B,这个过程只迁移内存不迁移磁盘;

[0131] 3、从中转服务器B直接迁移虚拟机到目标宿主机服务器C,最终完成虚拟机从源宿主服务器A迁移到目标宿主机服务器C;

[0132] 4、同时迁移完服务器A上的虚拟机后,就可以升级服务器A上的qemu。

[0133] 当然,这里所使用的升级qemu时通过上述第一种修改方式修改得到的系统代码。

[0134] 在本实施例中,通过上述两种修改方式对qemu的处理,就可以实现解析本地盘热迁移hang住,同时又可以在修改qemu后不关机升级qemu同时热迁移虚拟机。

[0135] 为了提高升级的优化,在本实施例中,可以同时使用上述两种方式进行升级,首先是在检测到需要进行升级时,基于服务器A本身的进程代码采用第一种修改方式修改代码,得到升级代码,然后根据升级代码对中转宿主机和目的宿主机进行升级,最后采用第二种修改方式实现数据的迁移,将服务器A上的数据迁移到服务器C上,再根据升级代码对服务器A进行升级。

[0136] 在实际应用中,在升级完成后,还可以将服务器C中的数据重复采用第二种修改方式将数据迁移会服务器A上运行。

[0137] 通过修改热迁移过程中对磁盘文件处理流程,减少raw driver判断或增加时间判断,可以有效的解决虚拟机使用过程中hang住问题,同时提供不关机升级qemu版本,可以有效解决当前qemu问题。

[0138] 通过这样修改迁移流程后,有如下好处:

[0139] 通过修改qemu的处理流程后,可以解决本地盘热迁移过程中可能存在的hang住问题。

[0140] 增加不关机升级qemu步骤后,可以不用关闭虚拟机直接升级qemu,同时热迁移虚拟机到其他宿主机上。

[0141] 本发明实施例可以在用户无感知的情况下解决问题,同时实现版本升级,大大提高产品竞争力。

[0142] 为了解决上述的问题,本发明实施例还提供了一种虚拟机升级装置,如图8所示,所述虚拟机升级装置包括:判断模块81、读取模块82、编译模块83、数据共享模块84、数据转移模块85和升级模块86,其中

[0143] 判断模块81,用于检测源宿主机的虚拟机程序的状态信息,并根据所述状态信息判断是否对虚拟机程序进行升级;

[0144] 读取模块82,用于在判断虚拟机程序需要升级时,接收所述虚拟机发起的升级请求,以及确定所述源宿主机正在运行的虚拟机程序的程序代码,其中,所述升级请求包括对所述程序代码的升级手段;

[0145] 编译模块83,用于根据所述程序代码的升级手段,对所述程序代码中被挂起的进程对应的代码进行修改,并对修改后的所述程序代码进行重新编译,得到新的系统程序;

[0146] 数据共享模块84,用于读取所述源宿主机中的运行内存数据和磁盘数据,将所述运行内存数据迁移至中转宿主主机上,并通过网络文件共享的方式将所述磁盘数据共享至中转宿主主机上;

[0147] 数据转移模块85,用于通过所述中转宿主机将所述运行内存数据迁移至目的宿主机上,以及所述中转宿主机通过远程访问的方式读取所述源宿主机上的磁盘数据发送至目的宿主机中,以实现所述源宿主机上的全部数据的迁移保存;

[0148] 升级模块86,用于关闭所述源宿主机,并根据所述系统程序对所述源宿主机上的虚拟系统进行升级处理。

[0149] 在本实施例中,所述状态信息为所述虚拟机程序当前运行的进程状态;

[0150] 所述判断模块用于判断所述虚拟机程序中的进程是否出现了挂起状态,其中,所述挂起状态为所述进程在同一程序节点上停留的时间不小于N倍该程序节点的预设执行时间,N为大于3的整数;若判断为挂起状态,则确定所述虚拟机程序需要进行升级。

[0151] 在本实施例中,所述编译模块包括代码检测单元、修改单元和编译单元,其中:

[0152] 所述代码检测单元用于检测出现挂机状态对应的程序节点在所述程序代码中位置,并根据所述位置获取所述程序节点所在的进程代码段;判断所述进程代码段是否属于所述程序代码的主进程程序;

[0153] 所述修改单元用于根据判断的结果和所述程序代码的升级手段选择对应的代码修改策略;

[0154] 所述编译单元用于根据所述代码修改策略对所述进程代码段进行升级,生成新的

系统程序。

[0155] 在本实施例中,所述修改单元用于,在判断所述进程代码段属于主进程程序时,选择的所述代码修改策略为增加跳转函数;或者用于在判断所述进程代码段不属于主进程程序时,选择的所述代码修改策略为删除代码段。

[0156] 在本实施例中,所述编译单元用于在所述代码修改策略为删除代码段时,将所述进程代码段从所述程序代码中删除,并重新编译生成所述系统程序;或者用于在所述代码修改策略为增加跳转函数时,在所述进程代码段中出现挂起状态的程序节点之后增加时间判断函数,通过所述时间判断函数对所述进程代码的挂机状态的停留时间进行判断,并重新编译生成所述系统程序。

[0157] 在本实施例中,若所述代码修改策略为增加跳转函数时,所述修改单元还用于在出现挂机状态的所述进程代码段之前的一个程序节点上设置程序跳转指针,并重新编译生成所述系统程序,其中所述程序跳转指针用于在所述时间判断函数确定所述停留时间超出预设执行时长时,将所述虚拟机从当前的系统程序跳转至其他的任务程序执行。

[0158] 在本实施例中,所述升级模块还用于根据所述系统程序分别对所述中转宿主机和目的宿主机中的虚拟机程序进行升级;

[0159] 所述数据迁移模块还用于在所述中转宿主机和目的宿主机升级完成后,执行将所述源宿主机上的所有数据通过所述中转宿主机迁移至所述目的宿主机上的步骤。

[0160] 基于本装置的执行功能和功能对应的执行流程与上述本发明实施例的虚拟机升级方法实施例说明内容相同的,因此本实施例对虚拟机升级装置的实施例内容不做过多赘述。

[0161] 此外,本发明实施例还提供了一种虚拟机升级设备,所述虚拟机升级设备包括:存储器、处理器以及存储在所述存储器上并可在所述处理器上运行的虚拟机升级程序,所述虚拟机升级程序被所述处理器执行时所实现的方法可参照本发明虚拟机升级方法的各个实施例,因此不再过多赘述。

[0162] 本发明还提供一种计算机可读存储介质。

[0163] 本实施例中,所述计算机可读存储介质上存储有虚拟机升级程序,所述虚拟机升级程序被处理器执行时所实现的方法可参照本发明虚拟机升级方法的各个实施例,因此不再过多赘述。

[0164] 通过以上的实施方式的描述,本领域的技术人员可以清楚地了解到上述实施例方法可借助软件加必需的通用硬件平台的方式来实现,当然也可以通过硬件,但很多情况下前者是更佳实施方式。基于这样的理解,本发明的技术方案本质上或者说对现有技术做出贡献的部分可以以软件产品的形式体现出来,该计算机软件产品存储在一个存储介质(如ROM/RAM)中,包括若干指令用以使得一台终端(可以是手机,计算机,服务器或者网络设备)执行本发明各个实施例所述的方法。

[0165] 上面结合附图对本发明的实施例进行了描述,但是本发明并不局限于上述的具体实施方式,上述的具体实施方式仅仅是示意性的,而不是限制性的,本领域的普通技术人员在本发明的启示下,在不脱离本发明宗旨和权利要求所保护的范围情况下,还可做出很多形式,凡是利用本发明说明书及附图内容所作的等效结构或等效流程变换,或直接或间接运用在其他相关的技术领域,这些均属于本发明的保护之内。

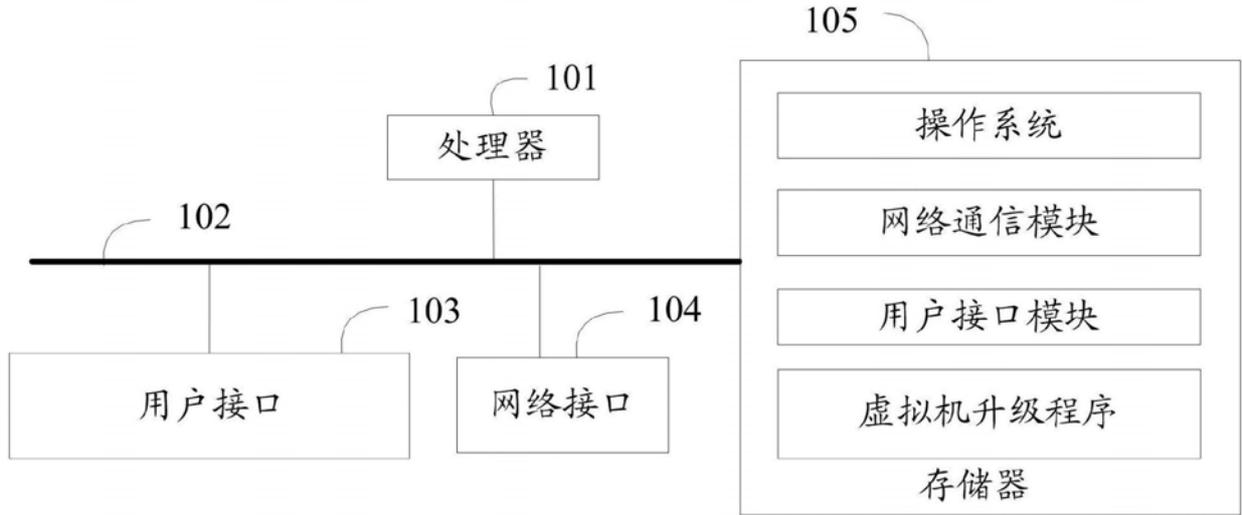


图1

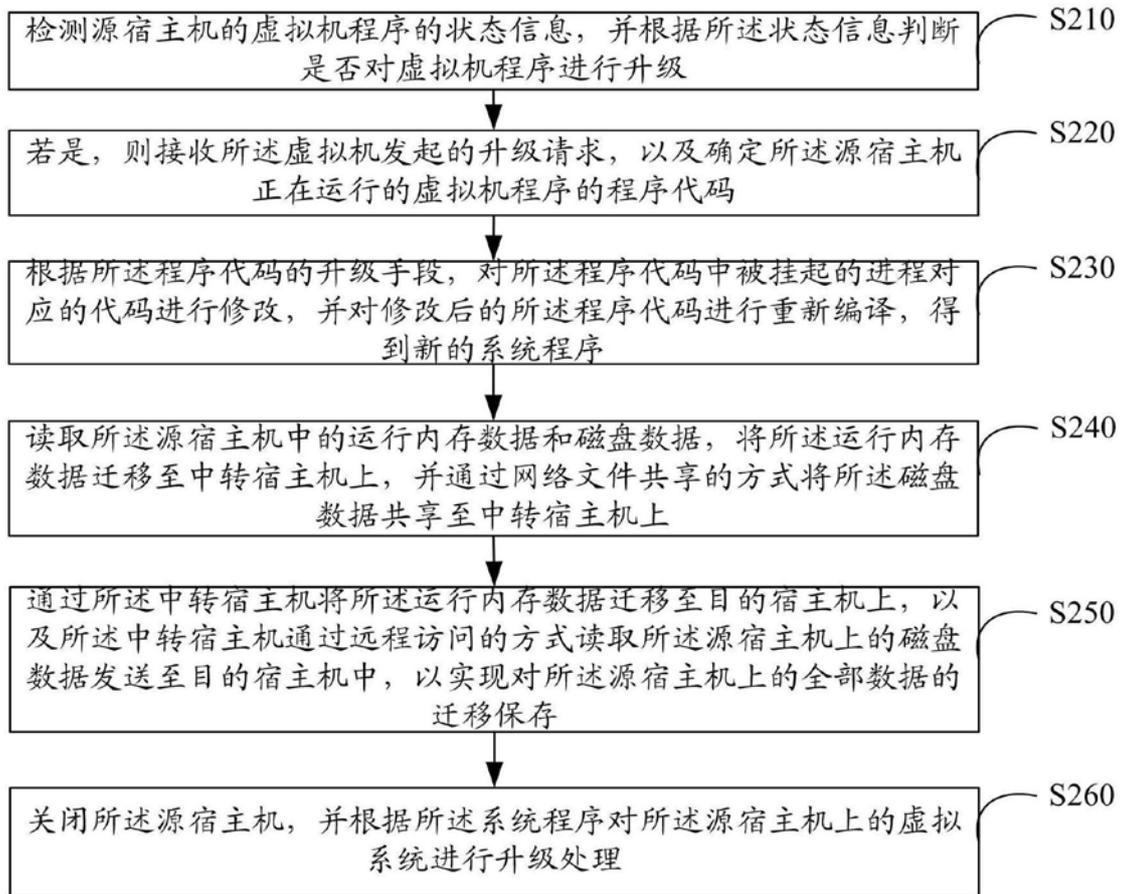


图2

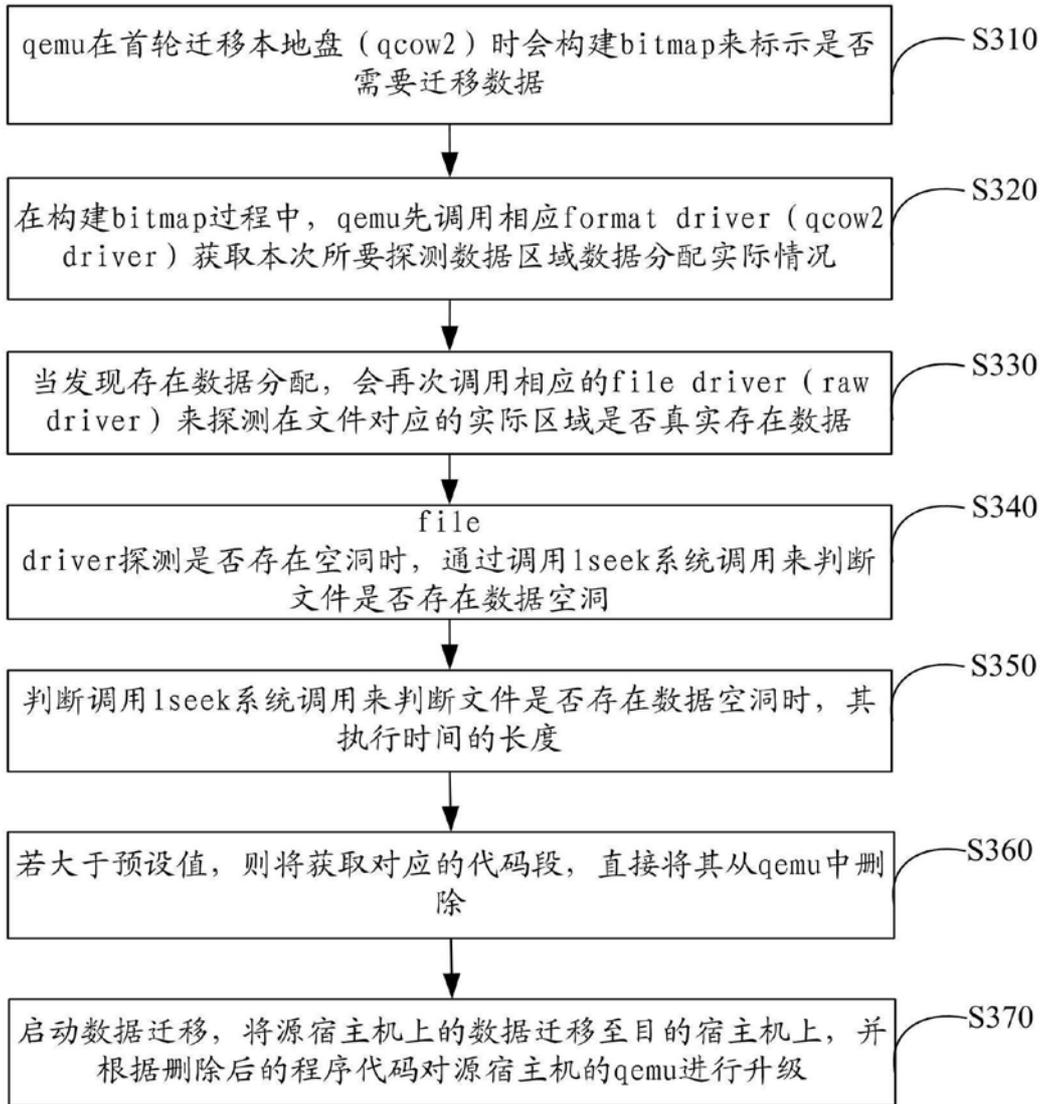


图3

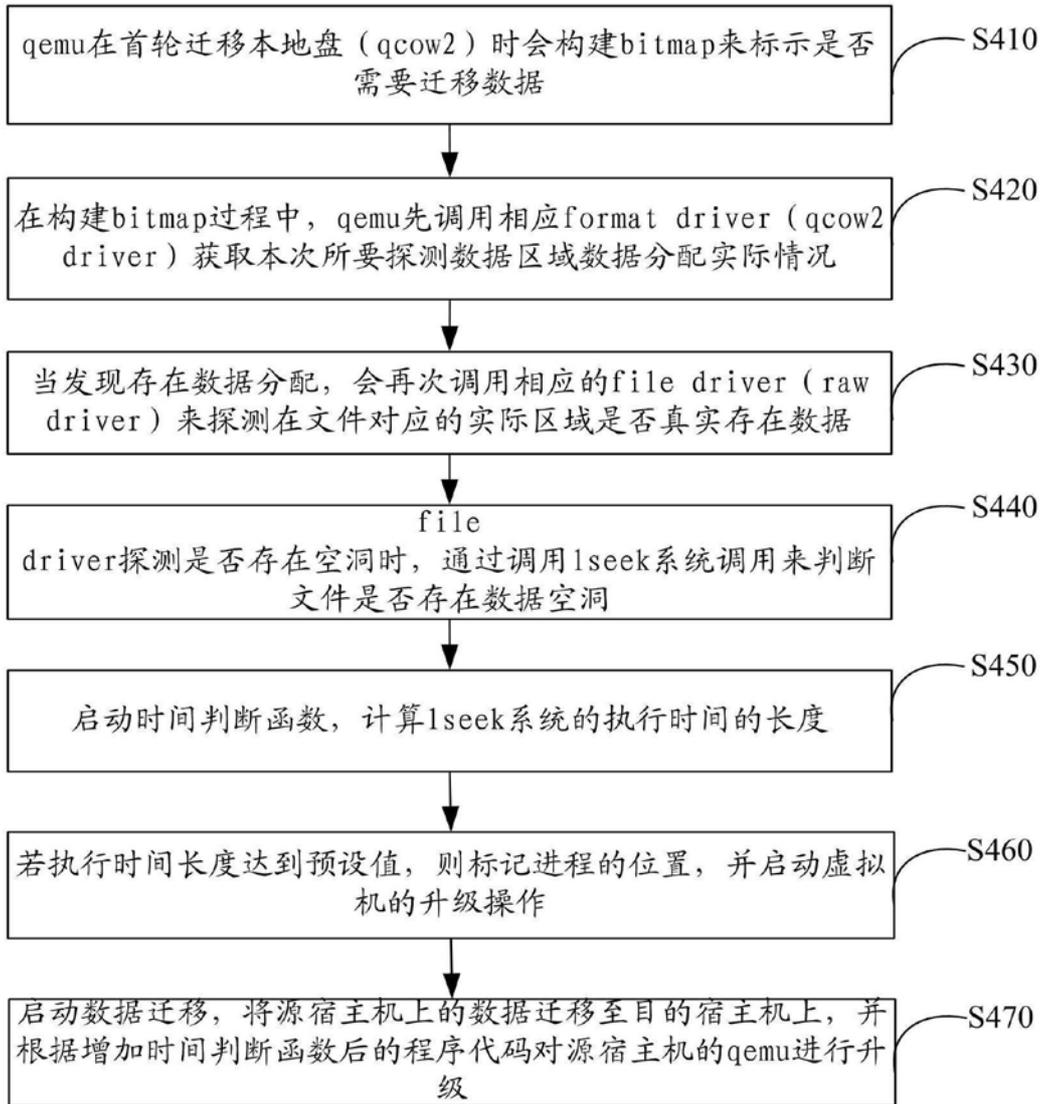


图4

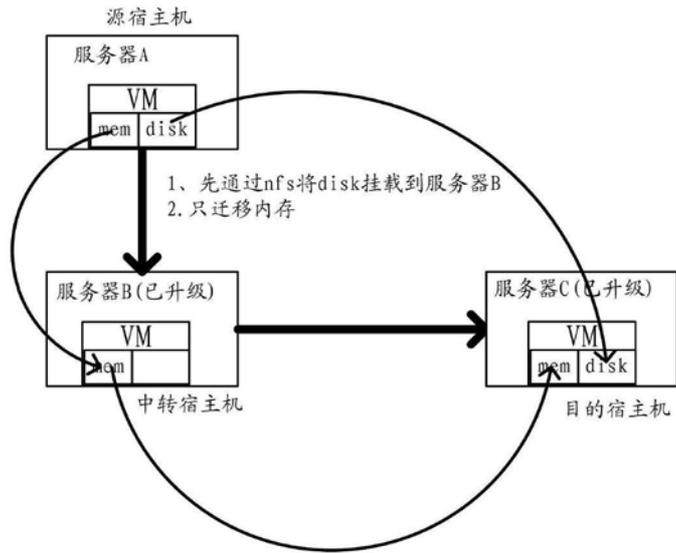


图5

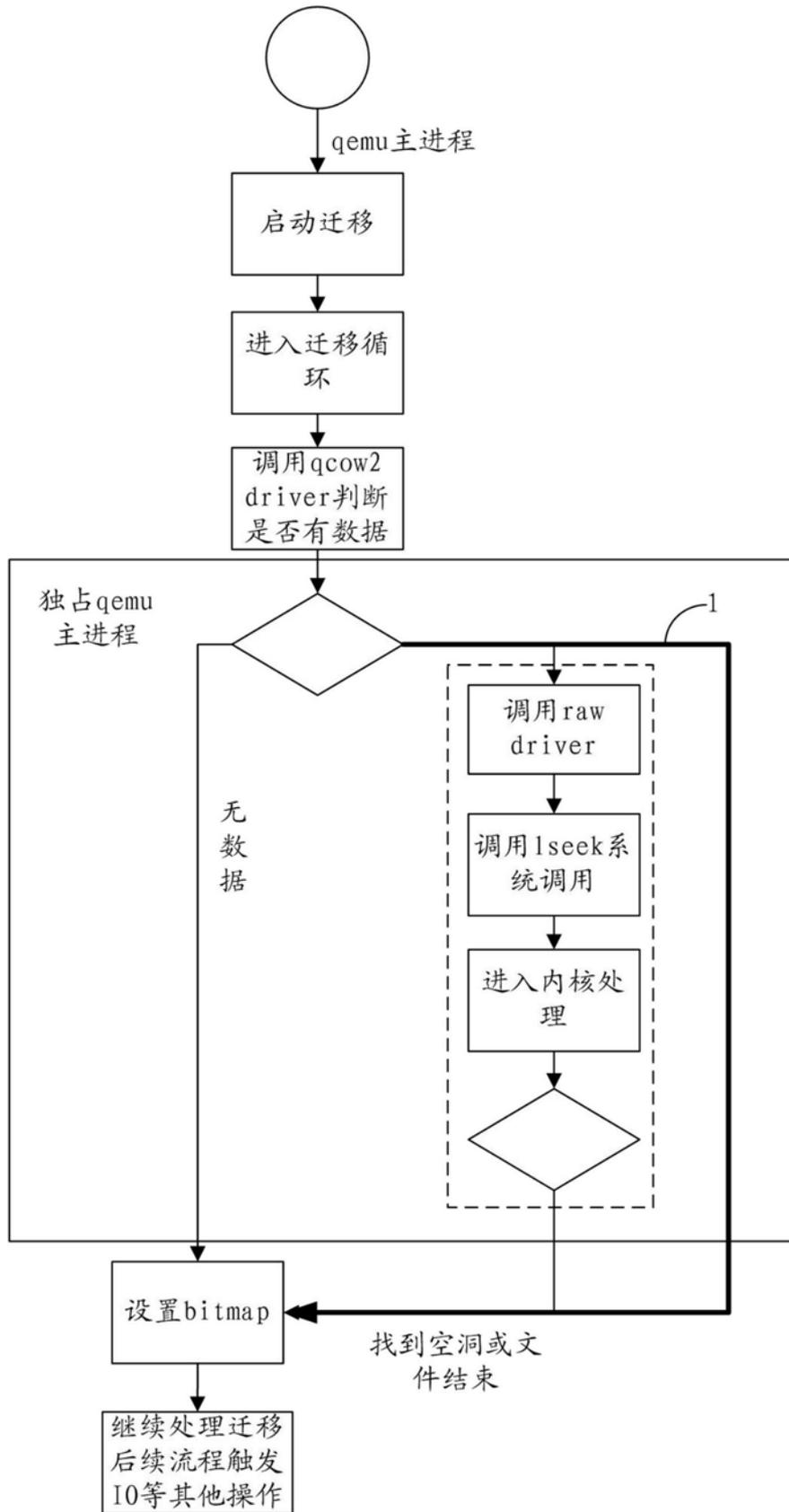


图6

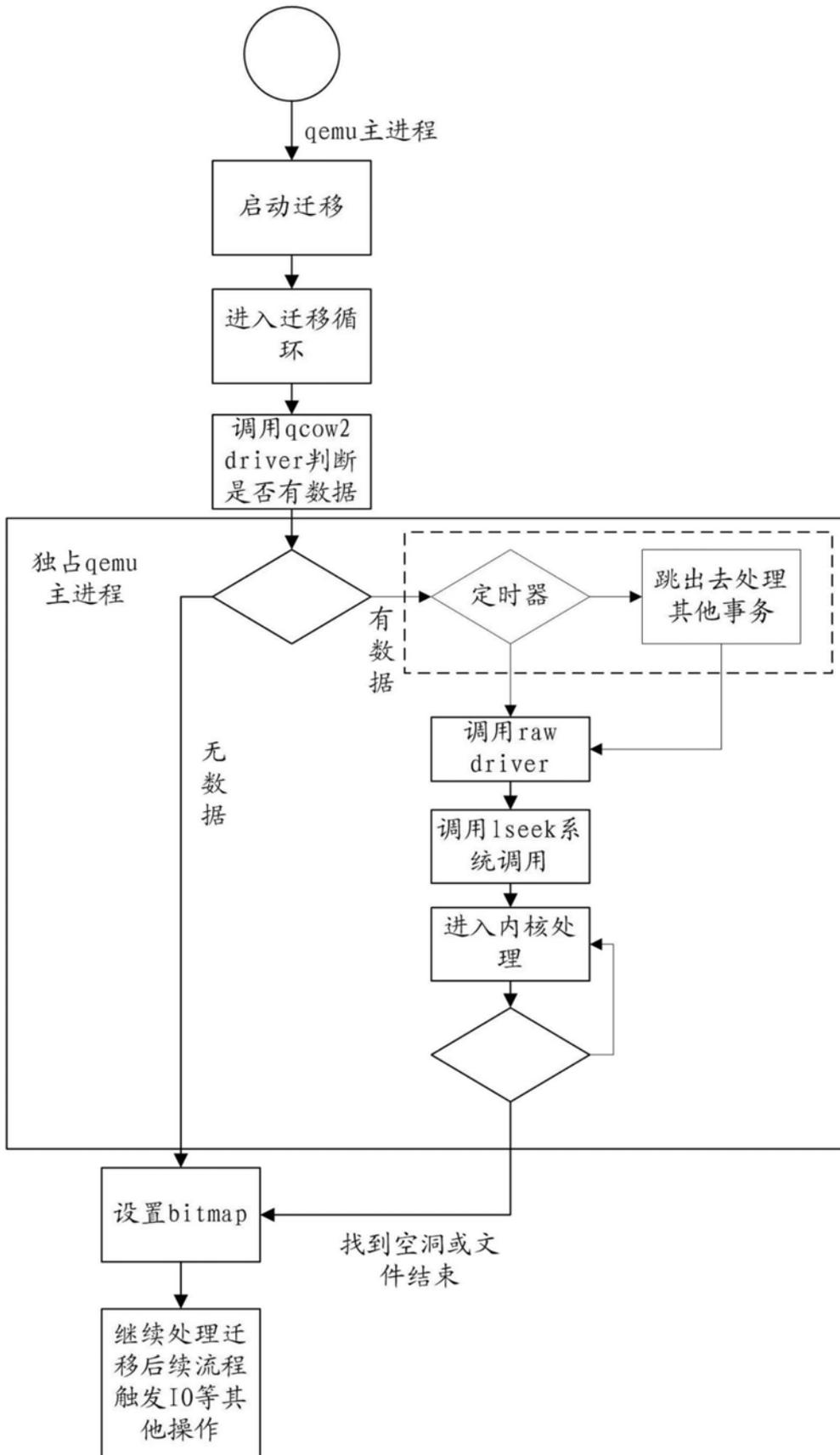


图7



图8