



(12) 发明专利申请

(10) 申请公布号 CN 118733190 A

(43) 申请公布日 2024.10.01

(21) 申请号 202310353270.X

(22) 申请日 2023.03.30

(71) 申请人 华为技术有限公司

地址 518129 广东省深圳市龙岗区坂田华为总部办公楼

(72) 发明人 蒋坤坤 王海滨 曾涛

(74) 专利代理机构 深圳市赛恩倍吉知识产权代理有限公司 44334

专利代理师 饶智彬 常云敏

(51) Int. Cl.

G06F 9/455 (2018.01)

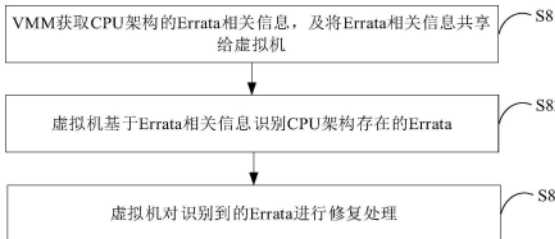
权利要求书3页 说明书21页 附图8页

(54) 发明名称

虚拟机处理勘误表的方法及相关设备

(57) 摘要

本申请提供了一种虚拟机处理勘误表的方法及相关设备,涉及计算机技术领域。虚拟机处理勘误表的方法,应用于电子设备,电子设备包括由至少一种物理CPU构成的CPU架构,电子设备运行有VM及用于管理VM的VMM,方法包括:VMM获取CPU架构的Errata相关信息,并将Errata相关信息共享给VM,Errata相关信息包括CPU类型信息与Timer Errata信息;VM基于Errata相关信息识别CPU架构存在的Errata;VM对识别到的Errata进行修复处理。本申请使得虚拟机可准确地处理CPU架构中存在的Errata,提升虚拟机运行的稳定性。



1. 一种虚拟机处理勘误表的方法,应用于电子设备,所述电子设备包括由至少一种物理中央处理单元CPU构成的CPU架构,其特征在于,所述电子设备运行有虚拟机VM及用于管理所述VM的虚拟机监视器VMM,所述方法包括:

所述VMM获取所述CPU架构的勘误表Errata相关信息,及将所述Errata相关信息共享给所述VM,所述Errata相关信息包括CPU类型信息与定时器勘误表Timer Errata信息;

所述VM基于所述Errata相关信息识别所述CPU架构存在的Errata;

所述VM对识别到的Errata进行修复处理。

2. 如权利要求1所述的虚拟机处理勘误表的方法,其特征在于,所述CPU类型信息包括所述CPU架构中的每种物理CPU的类型信息,所述Timer Errata信息包括第一Timer Errata信息与第二Timer Errata信息,所述VMM获取所述CPU架构的勘误表Errata相关信息,包括:

所述VMM遍历所述CPU架构中的每个物理CPU的系统寄存器,获取所述每种物理CPU的类型信息;

所述VMM将所述CPU类型信息与预设Timer Errata表进行比对,得到与所述CPU类型信息对应的所述第一Timer Errata信息;

所述VMM获取所述CPU架构中的固件的高级配置与电源接口ACPI表或设备树DT表,从所述ACPI表或所述DT表中读取Timer Errata相关信息字段,及基于所述Timer Errata相关信息字段解析得到所述第二Timer Errata信息。

3. 如权利要求1或2所述的虚拟机处理勘误表的方法,其特征在于,所述电子设备上配置有所述VM及所述VMM均具有访问权限的共享存储区,所述将所述Errata相关信息共享给所述VM,包括:

所述VMM将所述Errata相关信息写入至所述共享存储区;

所述VM从所述共享存储区读取所述Errata相关信息。

4. 如权利要求3所述的虚拟机处理勘误表的方法,其特征在于,所述VM基于所述Errata相关信息识别所述CPU架构存在的Errata,包括:

所述VM将所述CPU类型信息与预设CPU Errata表进行比对,得到所述CPU架构存在的CPU Errata;

所述VM基于所述Timer Errata信息,得到所述CPU架构存在的Timer Errata。

5. 如权利要求1至4中任意一项所述的虚拟机处理勘误表的方法,其特征在于,所述VM搭载客户操作系统guest OS,所述识别到的Errata包括静态管理类型Errata与动态管理类型Errata,所述VM对识别到的Errata进行修复处理,包括:

在所述VM的guest OS运行至第一Errata的Errata处理逻辑,且所述第一Errata为所述静态管理类型Errata,且确定所述CPU架构存在所述第一Errata时,所述VM执行所述第一Errata的Errata修复逻辑,以对所述第一Errata进行修复;

在所述VM的guest OS运行至第一Errata的Errata处理逻辑,且所述第一Errata为所述动态管理类型Errata,且确定所述CPU架构存在所述第一Errata时,所述VM执行所述第一Errata的Errata修复逻辑,以对所述第一Errata进行修复;

在所述VM的guest OS运行至第一Errata的Errata处理逻辑,且所述第一Errata为所述动态管理类型Errata,且确定所述CPU架构不存在所述第一Errata时,所述VM执行所述第一Errata的非修复逻辑。

6. 如权利要求5所述的虚拟机处理勘误表的方法,其特征在于,所述VM还包括动态Errata管理开关,所述在所述VM的guest OS运行至第一Errata的Errata处理逻辑,且所述第一Errata为所述动态管理类型Errata,且确定所述CPU架构存在所述第一Errata时,所述VM执行所述第一Errata的Errata修复逻辑,包括:

在所述VM的guest OS运行至第一Errata的Errata处理逻辑,且所述第一Errata为所述动态管理类型Errata,且确定所述CPU架构存在所述第一Errata,且所述动态Errata管理开关处于关闭状态时,所述VM执行所述第一Errata的Errata修复逻辑;

在所述VM的guest OS运行至第一Errata的Errata处理逻辑,且所述第一Errata为所述动态管理类型Errata,且确定所述CPU架构不存在所述第一Errata时,所述VM执行所述第一Errata的非修复逻辑,包括:

在所述VM的guest OS运行至第一Errata的Errata处理逻辑,且所述第一Errata为所述动态管理类型Errata,且确定所述CPU架构不存在所述第一Errata,且所述动态Errata管理开关处于关闭状态时,所述VM执行所述第一Errata的非修复逻辑。

7. 如权利要求1至4中任意一项所述的虚拟机处理勘误表的方法,其特征在于,所述VM还包括动态Errata管理开关,所述动态Errata管理开关处于开启状态,所述方法还包括:

所述VMM获取所述VM的虚拟中央处理单元VCPU线程所在的物理CPU类型信息,及将所述VCPU线程所在的物理CPU类型信息共享给所述VM,所述VM还可基于所述VCPU线程所在的物理CPU类型信息进行Errata修复处理。

8. 如权利要求7所述的虚拟机处理勘误表的方法,其特征在于,所述VM搭载guest OS,所述识别到的Errata包括静态管理类型Errata与动态管理类型Errata,所述VM对识别到的Errata进行修复处理,包括:

在所述VM的guest OS运行至第二Errata的Errata处理逻辑,且所述第二Errata为所述静态管理类型Errata,且确定所述CPU架构存在所述第二Errata时,所述VM执行所述第二Errata的Errata修复逻辑,以对所述第二Errata进行修复;

在所述VM的guest OS运行至第二Errata的Errata处理逻辑,且所述第二Errata为所述动态管理类型Errata,且基于所述VCPU线程所在的物理CPU类型信息确定所述VCPU线程所在的CPU存在所述第二Errata时,所述VM执行所述第二Errata的Errata修复逻辑,以对所述第二Errata进行修复;

在所述VM的guest OS运行至第二Errata的Errata处理逻辑,且所述第二Errata为所述动态管理类型Errata,且基于所述VCPU线程所在的物理CPU类型信息确定所述VCPU线程所在的CPU不存在所述第二Errata时,所述VM执行所述第二Errata的非修复逻辑。

9. 如权利要求8所述的虚拟机处理勘误表的方法,其特征在于,所述方法还包括:

所述VM将第二Errata处理上下文共享给所述VMM;

所述VMM基于所述第二Errata处理上下文管理所述VM的VCPU线程的调度。

10. 如权利要求9所述的虚拟机处理勘误表的方法,其特征在于,所述VMM基于所述第二Errata处理上下文管理所述VM的VCPU线程的调度,包括:

所述VMM基于所述第二Errata处理上下文,禁止所述VM的VCPU线程进行跨类型的CPU调度;或者

所述VMM基于所述第二Errata处理上下文,禁止所述VM的VCPU线程进行CPU调度。

11. 如权利要求10所述的虚拟机处理勘误表的方法,其特征在于,所述方法还包括:
在所述第二Errata处理完毕时,所述VM将第二Errata处理完毕上下文共享给所述VMM;
所述VMM基于所述第二Errata处理完毕上下文,恢复所述VM的VCPU线程的CPU调度权限。

12. 一种计算机可读存储介质,其特征在于,包括计算机指令,当所述计算机指令在电子设备上运行时,使得所述电子设备执行如权利要求1至权利要求11中任一项所述的虚拟机处理勘误表的方法。

13. 一种电子设备,其特征在于,所述电子设备包括处理器和存储器,所述存储器用于存储指令,所述处理器用于调用所述存储器中的指令,使得所述电子设备执行权利要求1至权利要求11中任一项所述的虚拟机处理勘误表的方法。

14. 一种计算机程序产品,其特征在于,包括计算机指令,当所述计算机指令在处理器上运行时,使得电子设备执行如权利要求1至权利要求11中任一项所述的虚拟机处理勘误表的方法。

虚拟机处理勘误表的方法及相关设备

技术领域

[0001] 本申请涉及计算机技术领域,尤其涉及一种虚拟机处理勘误表的方法及相关设备。

背景技术

[0002] 异构中央处理单元(central processing unit,CPU)架构是一种使用多种不同处理器的架构,例如big.LITTLE架构包括两种不同架构的CPU,其中“LITTLE”CPU旨在实现最高能效,“big”CPU旨在提供最高计算性能,两种CPU都可以访问相同的内存区域,因此工作负载可以在两种CPU间动态调度。

[0003] 对于异构CPU架构中不同类型的CPU而言,指令集、微架构、特性集等方面一般存在差异。面向这些差异,运行在异构CPU架构上的操作系统,往往需要进行一些架构上的兼容设计,比如屏蔽异构CPU间的差异,为整个操作系统提供统一的运行环境。不同类型的CPU的典型差异之一是勘误表(Errata)。CPU的指令中可能存在Errata(下文简称为CPU Errata),CPU的定时器组件中也可能存在Errata(下文简称Timer Errata)。操作系统可通过配置Errata管理系统对不同类型的CPU存在的Errata进行管理。

[0004] 基于异构CPU架构构建的虚拟机,虚拟机的虚拟CPU使用的固件由虚拟机监视器(virtual machine monitor,VMM)构建,虚拟机基于VMM创建的固件识别Timer Errata,而VMM创建的固件可能未添加相关Timer Errata描述字段,导致Timer Errata识别不全。虚拟机读取到的CPU类型信息与虚拟机启动时的虚拟CPU所在的物理CPU有关,进而可能导致识别到的CPU Errata不全,也可能导致识别到的Timer Errata不全。基于以上原因,虚拟机Errata管理系统无法获取到异构CPU架构中存在的所有Errata,进而导致虚拟机Errata管理系统无法顺利地处理异构CPU架构中存在的所有Errata,影响虚拟机内业务的正常运行。

发明内容

[0005] 有鉴于此,有必要提供一种虚拟机处理勘误表的方法,使得虚拟机可准确地处理CPU架构中存在的Errata,提升虚拟机运行的稳定性。

[0006] 本申请实施例第一方面公开了一种虚拟机处理勘误表的方法,应用于电子设备,电子设备包括由至少一个物理CPU构成的CPU架构,电子设备运行有虚拟机(virtual machine,VM)及用于管理VM的VMM,方法包括:VMM获取CPU架构的Errata相关信息,及将Errata相关信息共享给VM,Errata相关信息包括CPU类型信息与Timer Errata信息;VM基于Errata相关信息识别CPU架构存在的Errata;VM对识别到的Errata进行修复处理。

[0007] 采用上述技术方案,由运行在后端操作系统(即宿主机操作系统,host OS)的VMM维护CPU架构中的所有CPU类型信息及所有Timer Errata信息并共享给前端的VM,使得VM可识别到CPU架构中所有可能存在的Errata(例如CPU Errata及Timer Errata),可避免VM无法准确获取到异构CPU架构中存在的所有Errata,进而导致VM无法顺利地处理异构CPU架构中存在的Errata的问题,提升了VM内业务的运行稳定性。

[0008] 在一些实施例中,CPU类型信息包括CPU架构中的每种物理CPU的类型信息,Timer Errata信息包括第一Timer Errata信息与第二Timer Errata信息,VMM获取CPU架构的勘误表Errata相关信息,包括:VMM遍历CPU架构中的每个物理CPU的系统寄存器,获取每种物理CPU的类型信息;VMM将CPU类型信息与预设Timer Errata表进行比对,得到与CPU类型信息对应的所述第一Timer Errata信息;VMM获取CPU架构中的固件的高级配置与电源接口(advanced configuration and power interface ACPI)表或设备树(device tree,DT)表,从ACPI表或DT表中读取Timer Errata相关信息字段,及基于Timer Errata相关信息字段解析得到第二Timer Errata信息。

[0009] 采用上述技术方案,通过VMM在后端遍历CPU架构中的每个物理CPU的系统寄存器,从中读取每种物理CPU的类型信息,实现准确维护CPU架构中的所有CPU类型信息,通过VMM在后端对CPU架构中的固件的ACPI表或DT表进行解析,识别CPU架构可能存在的Timer Errata,及根据CPU类型信息补充识别CPU架构可能存在的Timer Errata,进而实现全面地收集CPU架构的Timer Errata信息,后端维护的CPU架构中的所有CPU类型信息及所有Timer Errata信息共享给前端的VM,使得VM可准确识别到CPU架构中所有可能存在的Errata,将VM的固件构建和Timer Errata的识别进行解耦(即不依赖VM的固件进行Timer Errata的识别),使得当出现新的Timer Errata时,无需修改VM的固件的构建代码。

[0010] 在一些实施例中,电子设备上配置有VM及VMM均具有访问权限的共享存储区,将Errata相关信息共享给VM,包括:VMM将Errata相关信息写入至共享存储区;VM从共享存储区读取Errata相关信息。

[0011] 采用上述技术方案,通过设置共享存储区实现VM与VMM之间进行通信,使得VMM在后端维护的信息可共享给前端的VM,VM在前端维护的信息可共享给后端的VMM,例如VMM可将共享给VM的信息写入至共享存储区,VM从共享存储区读取VMM写入的信息,VM可将共享给VMM的信息写入至共享存储区,VMM从共享存储区读取VM写入的信息。

[0012] 在一些实施例中,VM基于Errata相关信息识别CPU架构存在的Errata,包括:VM将CPU类型信息与预设CPU Errata表进行比对,得到CPU架构存在的CPU Errata;VM基于Timer Errata信息,得到CPU架构存在的Timer Errata。

[0013] 采用上述技术方案,VM可基于VMM共享的Errata相关信息准确地识别CPU架构存在的CPU Errata与Timer Errata,使得VM可顺利地处理异构CPU架构中存在的所有CPU Errata与Timer Errata,提升了VM内业务的运行稳定性。

[0014] 在一些实施例中,VM搭载客户操作系统(guest operating system,guest OS),识别到的Errata包括静态管理类型Errata与动态管理类型Errata,VM对识别到的Errata进行修复处理,包括:在VM的guest OS运行至第一Errata的Errata处理逻辑,且第一Errata为静态管理类型Errata,且确定CPU架构存在第一Errata时,VM执行第一Errata的Errata修复逻辑,以对第一Errata进行修复;在VM的guest OS运行至第一Errata的Errata处理逻辑,且第一Errata为动态管理类型Errata,且确定CPU架构存在第一Errata时,VM执行第一Errata的Errata修复逻辑,以对第一Errata进行修复;在VM的guest OS运行至第一Errata的Errata处理逻辑,且第一Errata为动态管理类型Errata,且确定CPU架构不存在第一Errata时,VM执行第一Errata的非修复逻辑。

[0015] 采用上述技术方案,Errata按照类型划分可包括CPU Errata、Timer Errata等等,

Errata还可根据管理方式进行划分,按照管理方式的不同可划分为两种类型:静态管理类型Errata与动态管理类型Errata,电子设备的操作系统(也可称为host OS)及VM的guest OS可包含各种Errata的Errata处理逻辑,以VM的guest OS为例,对于静态管理类型Errata,可以在VM的guest OS运行至某个Errata的Errata处理逻辑时,若CPU架构中存在该Errata,可直接执行Errata修复逻辑,对于动态管理类型Errata,VM的guest OS运行至某个Errata的Errata处理逻辑时,可基于CPU架构的CPU类型信息,确定是否执行Errata修复逻辑,例如,基于CPU类型信息确定CPU架构存在该Errata,执行Errata修复逻辑,对该Errata进行修复,若确定CPU架构不存在该Errata,执行非修复逻辑,即不对该Errata进行修复,避免执行冗余Errata修复逻辑,可提升VM的性能。

[0016] 在一些实施例中,VM还包括动态Errata管理开关,在VM的guest OS运行至第一Errata的Errata处理逻辑,且第一Errata为动态管理类型Errata,且确定CPU架构存在第一Errata时,VM执行第一Errata的Errata修复逻辑,包括:在VM的guest OS运行至第一Errata的Errata处理逻辑,且第一Errata为动态管理类型Errata,且确定CPU架构存在所述第一Errata,且动态Errata管理开关处于关闭状态时,VM执行第一Errata的Errata修复逻辑;在VM的guest OS运行至第一Errata的Errata处理逻辑,且第一Errata为动态管理类型Errata,且确定CPU架构不存在第一Errata时,VM执行第一Errata的非修复逻辑,包括:在VM的guest OS运行至第一Errata的Errata处理逻辑,且第一Errata为动态管理类型Errata,且确定CPU架构不存在第一Errata,且动态Errata管理开关处于关闭状态时,VM执行第一Errata的非修复逻辑。

[0017] 采用上述技术方案,可通过在VM中设置动态Errata管理开关来选择是否开启Errata动态处理机制,例如,VM可以响应于用户操作,将动态Errata管理开关置于关闭状态或者开启状态,若将动态Errata管理开关置于关闭状态,VM的guest OS运行至某个动态管理类型Errata的Errata处理逻辑时,可根据CPU类型信息确定是否执行Errata修复逻辑,若基于CPU类型信息确定CPU架构存在该动态管理类型Errata,执行Errata修复逻辑,基于CPU类型信息确定CPU架构不存在该动态管理类型Errata,执行非修复逻辑。

[0018] 在一些实施例中,VM还包括动态Errata管理开关,动态Errata管理开关处于开启状态,虚拟机处理勘误表的方法还包括:VMM获取VM的VCPU线程所在的物理CPU类型信息,及将VCPU线程所在的物理CPU类型信息共享给VM,VM还可基于VCPU线程所在的物理CPU类型信息进行Errata修复处理。

[0019] 采用上述技术方案,若将动态Errata管理开关置于开启状态,通过实时获取VCPU线程所在的物理CPU类型信息,基于VCPU线程所在的物理CPU类型信息确定是否执行Errata修复逻辑,避免执行冗余Errata修复逻辑,可进一步提升VM的性能。

[0020] 在一些实施例中,VM搭载guest OS,识别到的Errata包括静态管理类型Errata与动态管理类型Errata,VM对识别到的Errata进行修复处理,包括:在VM的guest OS运行至第二Errata的Errata处理逻辑,且第二Errata为静态管理类型Errata,且确定CPU架构存在所述第二Errata时,VM执行第二Errata的Errata修复逻辑,以对第二Errata进行修复;在VM的guest OS运行至第二Errata的Errata处理逻辑,且第二Errata为所述动态管理类型Errata,且基于VCPU线程所在的物理CPU类型信息确定VCPU线程所在的CPU存在第二Errata时,VM执行第二Errata的Errata修复逻辑,以对第二Errata进行修复;在VM的guest OS运行

至第二Errata的Errata处理逻辑,且第二Errata为动态管理类型Errata,且基于VCPU线程所在的物理CPU类型信息确定VCPU线程所在的CPU不存在第二Errata时,VM执行所述第二Errata的非修复逻辑。

[0021] 采用上述技术方案,Errata按照类型划分可包括CPU Errata、Timer Errata等等,Errata还可根据管理方式进行划分,按照管理方式的不同可划分为两种类型:静态管理类型Errata与动态管理类型Errata,电子设备的host OS及VM的guest OS可包含各种Errata的Errata处理逻辑,以VM的guest OS为例,对于静态管理类型Errata,在VM的guest OS运行至某个Errata的Errata处理逻辑时,若CPU架构中存在该Errata,可直接执行Errata修复逻辑,对于动态管理类型Errata,VM的guest OS运行至某个动态管理类型Errata的Errata处理逻辑时,可基于VCPU线程当前所在的物理CPU类型确定是否执行Errata修复逻辑,例如,通过获取VCPU线程当前所在的物理CPU类型,基于物理CPU类型确定该类型的物理CPU是否存在该动态管理类型Errata,若确定该类型的物理CPU中不存在该动态管理类型Errata,执行非修复逻辑,若确定该类型的物理CPU中存在该动态管理类型Errata,执行Errata修复逻辑,可避免执行冗余Errata修复逻辑,提升VM的性能。

[0022] 在一些实施例中,虚拟机处理勘误表的方法还包括:VM将第二Errata处理上下文共享给VMM;VMM基于第二Errata处理上下文管理VM的VCPU线程的调度。

[0023] 采用上述技术方案,通过在前端实时维护VCPU线程的Errata处理上下文,并共享给后端VMM,使得VMM在确定VCPU线程处于Errata处理逻辑中时,禁止VCPU线程进行跨类型的CPU调度,或者禁止VCPU线程进行CPU调度,保证了VCPU线程正确处理Errata,避免VCPU线程在处理某个Errata的非修改逻辑时,被调度到其他包含此Errata的CPU,导致Errata处理逻辑错误,影响VM的正常运行。

[0024] 在一些实施例中,VMM基于第二Errata处理上下文管理VM的VCPU线程的调度,包括:VMM基于第二Errata处理上下文,禁止VM的VCPU线程进行跨类型的CPU调度;或者VMM基于第二Errata处理上下文,禁止VM的VCPU线程进行CPU调度。

[0025] 采用上述技术方案,VMM在确定VCPU线程处于Errata处理逻辑中时,禁止VCPU线程进行跨类型的CPU调度,但可进行同类型的CPU调度,或者直接禁止VCPU线程进行CPU调度(即同类型的CPU调度亦不允许),保证了VCPU线程正确处理Errata。

[0026] 在一些实施例中,虚拟机处理勘误表的方法还包括:在第二Errata处理完毕时,VM将第二Errata处理完毕上下文共享给VMM;VMM基于第二Errata处理完毕上下文,恢复VM的VCPU线程的CPU调度权限。

[0027] 采用上述技术方案,在VCPU线程完成Errata处理之后,VMM确定VCPU线程未处于Errata处理逻辑中,恢复VM的VCPU线程的CPU自由调度权限,实现对VM进行全局CPU资源调度与管理,可充分利用CPU架构的CPU资源,充分发挥CPU架构的算力&功耗的平衡,灵活支撑VM的运维,提升VM上部署的应用体验。

[0028] 第二方面,本申请实施例提供一种计算机可读存储介质,包括计算机指令,当计算机指令在电子设备上运行时,使得电子设备执行如第一方面所述的虚拟机处理勘误表的方法。

[0029] 第三方面,本申请实施例提供一种电子设备,电子设备包括处理器和存储器,存储器用于存储指令,处理器用于调用存储器中的指令,使得电子设备执行如第一方面所述的

虚拟机处理勘误表的方法。

[0030] 第四方面,本申请实施例提供一种计算机程序产品,当计算机程序产品在电子设备(如计算机)上运行时,使得电子设备执行如第一方面所述的虚拟机处理勘误表的方法。

[0031] 第五方面,提供一种装置,所述装置具有实现上述第一方面所提供的方法中电子设备行为的功能。功能可以通过硬件实现,也可以通过硬件执行相应的软件实现。硬件或软件包括一个或多个与上述功能相对应的模块。

[0032] 可以理解地,上述提供的第二方面所述的计算机可读存储介质,第三方面所述的电子设备,第四方面所述的计算机程序产品,第五方面所述的装置均与上述第一方面的方法对应,因此,其所能达到的有益效果可参考上文所提供的对应的方法中的有益效果,此处不再赘述。

附图说明

[0033] 图1为本申请一实施例提供的异构CPU架构的虚拟机的CPU资源管理的系统架构示意图;

[0034] 图2为本申请一实施例提供的一种电子设备的结构示意图;

[0035] 图3为本申请一实施例提供的一种电子设备的软件结构示意图;

[0036] 图4为本申请一实施例提供的虚拟机处理勘误表的方法的应用架构示意图;

[0037] 图5a为本申请一实施例提供的部署在电子设备上的虚拟机进行勘误表处理的系统架构示意图;

[0038] 图5b为本申请另一实施例提供的部署在电子设备上的虚拟机进行勘误表处理的系统架构示意图;

[0039] 图6为本申请一实施例提供的部署在电子设备上的虚拟机与虚拟机监视器的交互示意图;

[0040] 图7为本申请另一实施例提供的部署在电子设备上的虚拟机与虚拟机监视器的交互示意图;

[0041] 图8为本申请一实施例提供的虚拟机处理勘误表的方法的步骤流程示意图。

具体实施方式

[0042] 需要说明的是,本申请中“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”,描述关联对象的关联关系,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的情况,其中A,B可以是单数或者复数。本申请的说明书和权利要求书及附图中的术语“第一”、“第二”、“第三”、“第四”等(如果存在)是用于区别类似的对象,而不是用于描述特定的顺序或先后次序。

[0043] 在本申请实施例中,“示例性的”或者“例如”等词用于表示作例子、例证或说明。本申请实施例中被描述为“示例性的”或者“例如”的任何实施例或设计方案不应被解释为比其它实施例或设计方案更优选或更具优势。确切而言,使用“示例性的”或者“例如”等词旨在以具体方式呈现相关概念。

[0044] 操作系统(Operation System,OS):管理计算机硬件与软件资源的计算机程序。

[0045] 进程:是一个具有一独立功能的程序关于某个数据集合的一次运行活动,进程是

操作系统动态执行的基本单元,在传统的操作系统中,进程既是基本的分配单元,也是基本的执行单元,进程可以被认为是正在运行的程序的实例。

[0046] 线程:是操作系统能够进行运算调度的最小单位,它被包含在进程之中,是进程中的实际运作单位。一条线程指的是进程中一个单一顺序的控制流,一个进程中可以并发多个线程,每条线程并行执行不同的任务。

[0047] 勘误表(Errata):是硬件设计完成后发现的某些特定功能在特定情况下会导致硬件偏离体系架构不能正常工作的缺陷,类似于软件程序中的bug。操作系统需要对Errata进行特殊处理,例如:执行特定的代码段或以特定方式配置硬件。

[0048] 虚拟化(virtualization):是一种资源管理技术,将主机的各种实体资源,如网络资源、内存及存储资源等,予以抽象、转换后呈现出来,打破主机的实体结构间的不可切割的障碍,使用户可以采用比原本的组态更好的方式来应用这些资源。通过虚拟化使用的资源称为虚拟化资源,虚拟化资源不受现有资源的架设方式、地域或物理组态所限制。一般地,虚拟化资源包括计算资源和存储资源。

[0049] 虚拟机(virtual machine,VM):是指通过软件在一台物理计算机上模拟出的一台或者多台虚拟计算机。这些虚拟机运行在完全隔离的环境中,就像真正的计算机那样进行工作。虚拟机上可以安装客户操作系统(guest operating system,guest OS),客户操作系统上运行有一个或多个应用。虚拟机还可访问网络资源。对于在虚拟机中运行的应用而言,就像是在真正的计算机中工作。

[0050] 宿主机(host)层:作为管理层,用以完成硬件资源的管理、分配;为虚拟机呈现虚拟硬件平台;实现虚拟机的调度和隔离等。在一些实现方式下,宿主机层包括宿主机操作系统(host operating system,host OS)和虚拟监控装置,例如,VMM或hypervisor,其中虚拟监控装置可部署在宿主机操作系统之内,也可以部署在宿主机操作系统之外。在另一些实现方式下,“宿主机层”还可以包括1个特权虚拟机(例如虚拟化架构Xen)。其中,虚拟硬件平台对其上运行的各个虚拟计算机提供各种硬件资源,如虚拟处理器、虚拟内存、虚拟磁盘、虚拟网卡等。虚拟计算机则运行在宿主机层为其准备的虚拟硬件平台上。本申请中有时将宿主机层简称为宿主机。

[0051] 硬件层:是指虚拟化环境运行的硬件平台。其中,硬件层可包括多种硬件,例如某物理计算机的硬件层可包括处理器和存储器,还可以包括中断控制器、网卡(network interface card,NIC)、输入/输出(input/output I/O)设备等。

[0052] 物理处理器:在本申请中指物理的处理单元,具体可以是一个最小处理单元,例如包含一个物理核,在某些实施例中也可以指包含多个物理核的处理单元。

[0053] 虚拟处理器:在虚拟化技术下,以共享或者分片方式提供给虚拟计算机使用的物理处理单元表示,例如,虚拟CPU(virtual central processing unit,vCPU)。一台虚拟计算机可以有一个或多个虚拟处理器为其服务,当存在多个虚拟处理器时,通常有一个虚拟处理器为主虚拟处理器,其他为从虚拟处理器。

[0054] 在宿主机上可以创建一个或多个虚拟机,宿主机的host OS及虚拟机的gust OS均可部署有Errata管理系统,Errata管理系统可对宿主机的CPU架构(同构CPU架构或异构CPU架构)中存在的Errata进行识别与修复处理,以保障宿主机的正常运行。例如,宿主机的host OS中的Errata管理系统,可在host OS运行过程中,对CPU架构中存在的Errata进行识

别与修复处理;虚拟机的gust OS中的Errata管理系统,可在gust OS运行过程中,对CPU架构中存在的Errata进行识别与修复处理,以保障虚拟机的正常运行。

[0055] 对于虚拟机而言,为了避免出现gust OS中的Errata管理系统在针对某一种类型CPU不存在的Errata进行非修复处理时,VCPU线程被调度到其他具有此Errata的CPU上,而导致Errata处理逻辑错误,影响虚拟机内业务的正常运行的问题,可采用VCPU绑核的方式来解决上述问题。如图1所示,假设电子设备包括两种类型的CPU(第一类型CPU与第二类型CPU),虚拟机VM1、VM3为同构核虚拟机,虚拟机VM2为混合核虚拟机,VMM针对各个虚拟机进行CPU资源管理。基于VCPU绑核的策略仅能支持区域CPU资源管理。即在虚拟机设计以及虚拟机初始化流程中,虚拟机的VCPU类型固定初始化成异构CPU架构中的某一类型CPU或某几种类型CPU;在整个虚拟机的生命周期内,以及对虚拟机的运维过程中,虚拟机的VCPU仅能在初始化资源管理的某一类型CPU内调度以及运维,无法支持全局CPU资源管理(无法在不同类型的CPU间进行调度),不能充分利用异构CPU架构的CPU资源,无法根据应用负载弹性全域调度,无法发挥异构CPU架构的算力&功耗的平衡,无法灵活支撑虚拟机的运维,影响虚拟机上部署的应用体验,且上述方案同样存在Errata识别不全的问题,导致虚拟机中的Errata管理系统无法顺利地处理异构CPU架构中存在的所有Errata。

[0056] 为解决上述技术问题,本申请一些实施例提供一种虚拟机处理勘误表的方法,通过在后端(宿主机OS)维护CPU架构中的所有CPU类型信息及所有Timer Errata信息并共享给前端的虚拟机,使得虚拟机的Errata管理系统可识别到所有可能存在的CPU Errata及所有可能存在的Timer Errata,避免出现Errata识别不全,导致虚拟机中的Errata管理系统无法顺利地处理异构CPU架构中存在的所有Errata的问题,且通过在后端实时维护更新VCPU线程所在物理CPU类型信息,在前端维护VCPU线程的Errata处理上下文,使得VCPU线程在处于Errata处理逻辑中时,禁止VCPU线程的跨类型的CPU调度,在Errata处理完成之后,恢复VCPU线程的CPU自由调度权限,实现对各个虚拟机进行全局CPU资源调度与管理,可充分利用异构CPU架构的CPU资源,充分发挥异构CPU架构的算力&功耗的平衡,灵活支撑虚拟机的运维,提升虚拟机上部署的应用体验。

[0057] 本申请实施例提供的虚拟机处理勘误表的方法,可应用于电子设备中,电子设备可以通过通信网络与其他电子设备或服务器进行通信。本申请的电子设备可以包括服务器或者终端设备。终端设备包括但不限于手机、可折叠电子设备、平板电脑、个人电脑(personal computer,PC)、膝上型计算机、手持计算机、笔记本电脑、超级移动个人计算机(ultra-mobile personal computer,UMPC)、上网本、蜂窝电话、个人数字助理(personal digital assistant,PDA)、增强现实(augmented reality,AR)设备、虚拟现实(virtual reality,VR)设备、人工智能(artificial intelligence,AI)设备、可穿戴式设备、智能家居设备、及智慧城市设备中的至少一种,本申请实施例对电子设备的具体类型不作特殊限制。通信网络可以是有线网络,也可以是无线网络。例如,通信网络可以是局域网(local area networks,LAN),也可以是广域网(wide area networks,WAN),例如互联网。当该通信网络为局域网时,示例性的,该通信网络可以是无线保真(wireless fidelity,Wi-Fi)热点网络、Wi-Fi P2P网络、蓝牙网络、zigbee网络或近场通信(near field communication,NFC)网络等近距离通信网络。当该通信网络为广域网时,示例性的,该通信网络可以是第三代移动通信技术(3rd-generation wireless telephone technology,3G)网络、第四代移

动通信技术(the 4th generation mobile communication technology,4G)网络、第五代移动通信技术(5th-generation mobile communication technology,5G)网络、未来演进的公共陆地移动网络(public land mobile network,PLMN)或因特网等。

[0058] 在一些实施例中,电子设备可以安装一个或多个APP(Application)。APP可以简称应用,为能够实现某项或多项特定功能的软件程序。例如,通讯类应用、视频类应用、音频类应用、图像拍摄类应用、云桌面类应用等等。其中,通信类应用,例如可以包括短信应用。图像拍摄类应用,例如可以包括拍摄应用(系统相机或第三方拍摄应用)。视频类应用,例如可以包括华为视频。音频类应用,例如可以包括华为音乐。以下实施例中提到的应用,可以是电子设备出厂时已安装的系统应用,也可以是用户在使用电子设备的过程中从网络下载或其他电子设备获取的第三方应用。

[0059] 电子设备包括但不限于搭载iOS®、安卓®(Android®)、Windows Mobile®、HarmonyOS®、或其他操作系统。本申请实施例以电子设备搭载Android®操作系统或HarmonyOS®操作系统为例进行说明。

[0060] 图2示意了一种电子设备10的结构示意图。

[0061] 电子设备10可以包括处理器110,外部存储器接口120,内部存储器121,天线1,天线2,移动通信模块130,无线通信模块140,音频模块150,传感器模块160,摄像模组170,显示屏180等。

[0062] 可以理解的是,本申请实施例示意的结构并不构成对电子设备10的具体限定。在本申请另一些实施例中,电子设备10可以包括比图示更多或更少的部件,或者组合某些部件,或者拆分某些部件,或者不同的部件布置。图示的部件可以以硬件,软件或软件和硬件的组合实现。

[0063] 处理器110可以包括一个或多个处理单元,例如:处理器110可以包括应用处理器(application processor,AP),调制解调处理器,图形处理器(graphics processing unit,GPU),图像信号处理器(image signal processor,ISP),控制器,视频编解码器,数字信号处理器(digital signal processor,DSP),基带处理器,和/或神经网络处理器(neural-network processing unit,NPU)等。其中,不同的处理单元可以是独立的器件,也可以集成在一个或多个处理器中。

[0064] 处理器110中还可以设置存储器,用于存储指令和数据。在一些实施例中,处理器110中的存储器可以为高速缓冲存储器。所述存储器可以保存处理器110用过或使用频率较高的指令或数据。如果处理器110需要使用所述指令或数据,可从所述存储器中直接调用。避免了重复存取,减少了处理器110的等待时间,因而提高了系统的效率。

[0065] 在一些实施例中,处理器110可以包括一个或多个接口。接口可以包括内部集成电路(inter-integrated circuit,I2C)接口,集成电路内置音频(inter-integrated circuit sound,I2S)接口,脉冲编码调制(pulse code modulation,PCM)接口,通用异步收发传输器(universal asynchronous receiver/transmitter,UART)接口,移动产业处理器接口(mobile industry processor interface,MIPI),通用输入输出(general-purpose input/output,GPIO)接口,用户标识模块(subscriber identity module,SIM)接口,和/或通用串行总线(universal serial bus,USB)接口等。处理器110可以通过以上至少一种接

口连接音频模块、无线通信模块、显示器、摄像头等模块。

[0066] 可以理解的是,本申请实施例示意的各模块间的接口连接关系,只是示意性说明,并不构成对电子设备10的结构限定。在本申请另一些实施例中,电子设备10也可以采用上述实施例中不同的接口连接方式,或多种接口连接方式的组合。

[0067] 电子设备10的无线通信功能可以通过天线1,天线2,移动通信模块130,无线通信模块140,调制解调处理器以及基带处理器等实现。

[0068] 移动通信模块130可以提供应用在电子设备10上的包括2G/3G/4G/5G等无线通信的解决方案。移动通信模块130可以包括至少一个滤波器,开关,功率放大器,低噪声放大器(low noise amplifier,LNA)等。在一些实施例中,移动通信模块130的至少部分功能模块可以被设置于处理器110中。在一些实施例中,移动通信模块130的至少部分功能模块可以与处理器110的至少部分模块被设置在同一个器件中。

[0069] 无线通信模块140可以提供应用在电子设备10上的包括无线局域网(wireless local area networks,WLAN)(如无线保真(wireless fidelity,Wi-Fi)网络),蓝牙(blueetooth,BT),蓝牙低功耗(blueetooth low energy,BLE),超宽带(ultra wide band,UWB),全球导航卫星系统(global navigation satellite system,GNSS),调频(frequency modulation,FM),近距离无线通信技术(near field communication,NFC),红外技术(infrared,IR)等无线通信的解决方案。无线通信模块140可以是集成至少一个通信处理模块的一个或多个器件。

[0070] 在一些实施例中,电子设备10可以通过无线通信技术与网络和其他电子设备通信。所述无线通信技术可以包括全球移动通讯系统(global system for mobile communications,GSM),通用分组无线服务(general packet radio service,GPRS),码分多址接入(code division multiple access,CDMA),宽带码分多址(wideband code division multiple access,WCDMA),时分码分多址(time-division code division multiple access,TD-SCDMA),长期演进(long term evolution,LTE),BT,GNSS,WLAN,NFC,FM,和/或IR技术等。所述GNSS可以包括全球卫星定位系统(global positioning system,GPS),全球导航卫星系统(global navigation satellite system,GLONASS),北斗卫星导航系统(beidou navigation satellite system,BDS),准天顶卫星系统(quasi-zenith satellite system,QZSS)和/或星基增强系统(satellite based augmentation systems,SBAS)。

[0071] 电子设备10可以通过GPU,显示屏180,以及应用处理器等实现显示功能。GPU为图像处理的微处理器,连接显示屏180和应用处理器。GPU用于执行数学和几何计算,用于图形渲染。处理器110可包括一个或多个GPU,其执行程序指令以生成或改变显示信息。

[0072] 摄像模组170包括摄像头。显示屏180用于显示图像,视频等,显示屏180可以为触控屏。显示屏180包括显示面板。显示面板可以采用液晶显示屏(liquid crystal display,LCD),有机发光二极管(organic light-emitting diode,OLED),有源矩阵有机发光二极体或主动矩阵有机发光二极体(active-matrix organic light emitting diode的,AMOLED),柔性发光二极管(flexible light-emitting diode,FLED),Miniled,MicroLed, Micro-oLed,量子点发光二极管(quantum dot light emitting diodes,QLED)等。在一些实施例中,电子设备10可以包括1个或多个显示屏180。

[0073] 外部存储器接口120可以用于连接外部存储卡,例如Micro SD卡,实现扩展电子设备10的存储能力。外部存储卡通过外部存储器接口120与处理器110通信,实现数据存储功能。

[0074] 内部存储器121可以用于存储计算机可执行程序代码,所述可执行程序代码包括指令。内部存储器121可以包括存储程序区和存储数据区。其中,存储程序区可存储操作系统,至少一个功能所需的应用程序(比如声音播放功能,图像播放功能等)等。存储数据区可存储电子设备10使用过程中所创建的数据等。此外,内部存储器121可以包括高速随机存取存储器,还可以包括非易失性存储器,例如至少一个磁盘存储器件,闪存器件,通用闪存存储器(universal flash storage,UFS)等。处理器110通过运行存储在内部存储器121的指令,和/或存储在设置于处理器中的存储器的指令,执行电子设备10的各种功能方法或数据处理。

[0075] 音频模块150用于将数字音频信息转换成模拟音频信号输出,也用于将模拟音频输入转换为数字音频信号。音频模块150还可以用于对音频信号编码和解码。在一些实施例中,音频模块150可以设置于处理器110中,或将音频模块150的部分功能模块设置于处理器110中。

[0076] 电子设备10的软件系统可以采用分层架构,事件驱动架构,微核架构,微服务架构,或云架构。本申请实施例以分层架构的Android系统为例,示例性说明电子设备10的软件结构。

[0077] 图3是本申请一实施例的电子设备的软件结构框图。

[0078] 分层架构将软件分成若干个层,每一层都有清晰的角色和分工。层与层之间通过软件接口通信。在一些实施例中,将Android系统分为五层,从上至下分别为应用程序层,应用程序框架层,安卓运行时(Android runtime,ART)和原生C/C++库,硬件抽象层(Hardware Abstract Layer,HAL)以及内核层。

[0079] 应用程序层可以包括一系列应用程序包。

[0080] 如图3所示,应用程序包可以包括相机,图库,日历,通话,地图,导航,WLAN,蓝牙,音乐,视频,短信息等应用程序。

[0081] 应用程序框架层为应用程序层的应用程序提供应用编程接口(application programming interface,API)和编程框架。应用程序框架层包括一些预先定义的函数。

[0082] 如图3所示,应用程序框架层可以包括窗口管理器,内容提供者,视图系统,资源管理器,通知管理器,活动管理器,输入管理等。

[0083] 窗口管理器提供窗口管理服务(Window Manager Service,WMS),WMS可以用于窗口管理、窗口动画管理、surface管理以及作为输入系统的中转站。

[0084] 内容提供者用来存放和获取数据,并使这些数据可以被应用程序访问。所述数据可以包括视频,图像,音频,拨打和接听的电话,浏览历史和书签,电话簿等。

[0085] 视图系统包括可视控件,例如显示文字的控件,显示图片的控件等。视图系统可用于构建应用程序。显示界面可以由一个或多个视图组成的。例如,包括短信通知图标的显示界面,可以包括显示文字的视图以及显示图片的视图。

[0086] 资源管理器为应用程序提供各种资源,比如本地化字符串,图标,图片,布局文件,视频文件等等。

[0087] 通知管理器使应用程序可以在状态栏中显示通知信息,可以用于传达告知类型的消息,可以短暂停留后自动消失,无需用户交互。比如通知管理器被用于告知下载完成,消息提醒等。通知管理器还可以是以图表或者滚动条文本形式出现在系统顶部状态栏的通知,例如后台运行的应用程序的通知,还可以是以对话框形式出现在屏幕上的通知。例如在状态栏提示文本信息,发出提示音,电子设备振动,指示灯闪烁等。

[0088] 活动管理器可以提供活动管理服务(Activity Manager Service,AMS),AMS可以用于系统组件(例如活动、服务、内容提供者、广播接收器)的启动、切换、调度以及应用进程的管理和调度工作。

[0089] 输入管理器可以提供输入管理服务(Input Manager Service,IMS),IMS可以用于管理系统的输入,例如触摸屏输入、按键输入、传感器输入等。IMS从输入设备节点取出事件,通过和WMS的交互,将事件分配至合适的窗口。

[0090] 安卓运行时包括核心库和安卓运行时。安卓运行时负责将源代码转换为机器码。安卓运行时主要包括采用提前(ahead or time,AOT)编译技术和及时(just in time,JIT)编译技术。

[0091] 核心库主要用于提供基本的Java类库的功能,例如基础数据结构、数学、IO、工具、数据库、网络等库。核心库为用户进行安卓应用开发提供了API。

[0092] 原生C/C++库可以包括多个功能模块。例如:表面管理器(surface manager),媒体框架(Media Framework),libc,OpenGL ES,SQLite,Webkit等。

[0093] 其中,表面管理器用于对显示子系统进行管理,并且为多个应用程序提供了2D和3D图层的融合。媒体框架支持多种常用的音频,视频格式回放和录制,以及静态图像文件等。媒体库可以支持多种音视频编码格式,例如:MPEG4,H.264,MP3,AAC,AMR,JPG,PNG等。OpenGL ES提供应用程序中2D图形和3D图形的绘制和操作。SQLite为电子设备10的应用程序提供轻量级关系型数据库。

[0094] 硬件抽象层运行于用户空间(user space),对内核层驱动进行封装,向上层提供调用接口。

[0095] 内核层是硬件和软件之间的层。内核层至少包含显示驱动,摄像头驱动,音频驱动,传感器驱动。

[0096] 下面结合图4,示例性的介绍虚拟机处理勘误表的方法的应用架构示意图。

[0097] 在该实施例中,应用架构可包括硬件层与软件层。硬件层可包括一种或多种硬件资源,图4以硬件层包括CPU资源为例进行说明。假设,电子设备10包括两种架构的CPU(为便于区分,以下称为第一类型CPU与第二类型CPU),第一类型CPU可包括一个或多个物理核core1,第二类型CPU可包括一个或多个物理核core2。电子设备10可以是终端设备,或者服务器,本申请实施例对此不作限定。

[0098] 软件层可包括一个或多个虚拟机及对该一个或多个虚拟机进行管理的VMM,VMM可以是第一类VMM(Type-1 VMM),第二类VMM(Type-2 VMM)等,本申请对此不作限定。图4以软件层包括多个虚拟机为例。本申请实施例所称的模块/单元可以是指能够完成特定功能的一系列计算机程序指令段,也可以是计算机程序指令段与硬件配合形成的功能模块,模块的划分为一种逻辑功能划分,实际实现时可以有另外的划分方式,本申请对此不作限定。

[0099] VMM可包括虚拟化Errata管理后端模块101及资源管理模块102。虚拟化Errata管

理后端模块101用于收集CPU类型信息与Timer Errata信息,CPU类型信息与Timer Errata信息可以数组、链表、位图等方式进行呈现与保存,本申请对此不作限定。例如,虚拟化Errata管理后端模块101可通过遍历电子设备10中的所有CPU,通过读取每个CPU的系统寄存器来实现收集CPU类型信息;虚拟化Errata管理后端模块101还可通过读取Timer Errata相关信息字段识别存在的Timer Errata及根据CPU类型信息识别存在的Timer Errata,进而实现全面地收集Timer Errata信息。例如,对于高级配置与电源接口(advanced configuration and power interface,ACPI)类型的固件(宿主机的CPU架构使用的固件,在电子设备出厂时已由设备厂商配置好),虚拟化Errata管理后端模块101可通过解析ACPI表,从中获取Timer Errata相关信息字段,进而识别出存在的Timer Errata;对于设备树(device tree,DT)类型的固件(宿主机的CPU架构使用的固件),虚拟化Errata管理后端模块101可通过解析DT表,从中获取Timer Errata相关信息字段,进而识别出存在的Timer Errata。

[0100] 资源管理模块102用于对各个虚拟机进行资源调度与管理,例如,对各个虚拟机进行CPU资源调度与管理。

[0101] 每个虚拟机均可包括虚拟化Errata管理前端模块103及Errata管理系统104。虚拟化Errata管理前端模块103用于获取虚拟化Errata管理后端模块101收集到的CPU类型信息与Timer Errata信息,及基于CPU类型信息与Timer Errata信息识别异构CPU架构中存在的Errata(CPU Errata与Timer Errata)。Errata管理系统104用于在识别异构CPU架构中或者某个类型的CPU中存在的Errata时,调用虚拟化Errata管理前端模块103进行Errata识别,Errata管理系统104还用于对Errata进行修复处理,或者非修复处理。

[0102] 在一些实施例中,对Errata进行修复处理可以是指执行Errata修复逻辑,进行Errata修复,对Errata进行非修复处理可以是指执行非修复逻辑(对该Errata不进行修复)。

[0103] 例如,一个CPU Errata为对CPU中的高速缓冲存储器(cache)存储的数据进行擦除时,存在Errata,执行一次擦除指令可能会存在数据擦除不干净的情形,该CPU Errata的修复逻辑为执行二次擦除指令。

[0104] 例如,一个Timer Errata为从Timer读取某个count的数据时存在Errata,进行一次读取可能会存在数据读取不准确的情形,该CPU Errata的修复逻辑为进行多次读取(比如读取三次)。

[0105] 在一些实施例中,虚拟化Errata管理前端模块103也可以集成至Errata管理系统104,本申请对此不作限定。Errata修复逻辑与非修复逻辑可以是一系列计算机程序指令段,本申请对计算机程序指令段的具体代码内容不作限定。

[0106] 在一些实施例中,虚拟化Errata管理后端模块101还用于实时维护虚拟机的VCPU线程所在的物理CPU类型信息,虚拟化Errata管理前端模块103可获取虚拟化Errata管理后端模块101维护的虚拟机的VCPU线程当前所在的物理CPU类型信息。Errata管理系统104在对某个Errata进行处理之前,可基于VCPU线程所在的物理CPU类型信息确定是执行Errata修复逻辑,还是非修复逻辑。例如,Errata管理系统104在对第一Errata进行处理之前,基于VCPU线程所在的物理CPU类型信息确定该类型的物理CPU存在的所有Errata,若该类型的物理CPU存在第一Errata,执行Errata修复逻辑(对第一Errata进行修复),若该类型的物理

CPU不存在第一Errata,执行非修复逻辑(即对第一Errata不进行修复)。

[0107] 在一些实施例中,在Errata管理系统104对某个Errata进行处理的过程中,虚拟化Errata管理前端模块103还用于维护Errata处理上下文。虚拟化Errata管理后端模块101可获取虚拟化Errata管理前端模块103维护的Errata处理上下文,对虚拟机的VCPU线程的调度进行管理。例如,虚拟化Errata管理后端模块101基于Errata处理上下文确定虚拟机处于Errata处理逻辑中,禁止虚拟机的VCPU线程进行跨类型的物理CPU调度。

[0108] 如图5a所示,以软件层包括一个虚拟机及对该虚拟机进行管理的VMM为例进行说明。电子设备10的CPU架构可以是异构CPU架构或者同构CPU架构。

[0109] 虚拟化Errata管理后端模块101可包括CPU类型收集单元1011及Timer Errata收集单元1012。CPU类型收集单元1011用于收集CPU类型信息,例如CPU类型收集单元1011可通过遍历电子设备10中的所有CPU,读取每个CPU的系统寄存器来实现收集CPU类型信息。Timer Errata收集单元1012用于收集Timer Errata信息,例如Timer Errata收集单元1012可通过解析CPU架构的固件的ACPI表或DT表,读取ACPI表或DT表中的Timer Errata相关信息字段,识别出存在的Timer Errata。Timer Errata收集单元1012还根据CPU类型信息识别存在的Timer Errata,例如基于CPU类型信息在CPU Timer驱动中的Timer Errata表(Timer Errata表可存储有各个CPU类型对应的Timer Errata)中进行遍历,识别出存在的Timer Errata。

[0110] 虚拟化Errata管理前端模块103可通过信息共享的通道获取CPU类型收集单元1011及Timer Errata收集单元1012收集到的CPU类型信息与Timer Errata信息。信息共享的通道可以是指实现虚拟化Errata管理前端模块103与虚拟化Errata管理后端模块101进行信息交互的通道。

[0111] 在一些实施例中,可以基于半虚拟化(para-virtualization,PV)技术实现构建虚拟化Errata管理前端模块103与虚拟化Errata管理后端模块101之间进行信息共享的通道。例如,可基于共享内存、Hypervisor Call (HVC)等实现构建虚拟化Errata管理前端模块103与虚拟化Errata管理后端模块101之间进行信息共享的通道。以共享内存为例,虚拟化Errata管理前端模块103可将共享给虚拟化Errata管理后端模块101的信息写入至共享内存,虚拟化Errata管理后端模块101从共享内存读取虚拟化Errata管理前端模块103共享的信息,同理,虚拟化Errata管理后端模块101可将共享给虚拟化Errata管理前端模块103的信息写入至共享内存,虚拟化Errata管理前端模块103从共享内存读取虚拟化Errata管理后端模块101共享的信息。

[0112] 虚拟化Errata管理前端模块103可基于CPU类型信息与Timer Errata信息识别电子设备10的CPU架构中存在的Errata。Errata管理系统104可在需要进行Errata识别时,调用虚拟化Errata管理前端模块103进行Errata识别。

[0113] 虚拟机的guest OS运行至某个Errata的Errata处理逻辑时,Errata管理系统104还可根据CPU类型信息确定是对该Errata进行修复处理,或者非修复处理。例如,若根据CPU类型信息确定CPU架构存在该Errata,对该Errata进行修复处理(执行Errata修复逻辑),若根据CPU类型信息确定CPU架构不存在该Errata,对该Errata进行非修复处理(执行非修复逻辑)。

[0114] 上述方案通过在后端维护CPU架构中的所有CPU类型信息及所有Timer Errata信

息并共享给前端的虚拟机,使得虚拟机的Errata管理系统可识别到所有可能存在的CPU Errata及所有可能存在的Timer Errata,可避免虚拟机Errata管理系统无法获取到异构CPU架构中存在的所有Errata,进而导致虚拟机Errata管理系统无法顺利地处理异构CPU架构中存在的所有Errata的问题,提升了虚拟机内业务的运行稳定性,同时将虚拟机的固件构建和Timer Errata的识别进行解耦(不依赖虚拟机的固件进行Timer Errata的识别),使得当出现新的Timer Errata时,无需修改虚拟机的固件的构建代码。

[0115] 如图5b所示,该应用架构提供一种Errata动态处理机制,通过使用动态Errata管理开关来选择是否开启Errata动态处理机制。例如,虚拟机可以响应于用户操作,将动态Errata管理开关置于关闭状态或者开启状态。电子设备10的CPU架构以异构CPU架构为例进行说明。

[0116] 虚拟化Errata管理后端模块101可包括CPU类型收集单元1011、Timer Errata收集单元1012、VCPU类型维护单元1013及VCPU调度管理单元1014。CPU类型收集单元1011与Timer Errata收集单元1012的功能在上述实施例中已有介绍,在此不再赘述。VCPU类型维护单元1013用于维护虚拟机的VCPU线程实时所在的物理CPU类型信息,当VCPU线程调度到其他CPU时,VCPU类型维护单元1013可实时地进行VCPU线程所在的物理CPU类型信息的更新。VCPU调度管理单元1014用于控制VCPU线程的调度,例如当确定VCPU处于Errata处理逻辑中时,禁止VCPU线程进行跨类型的CPU调度。

[0117] 若将动态Errata管理开关置于关闭状态,虚拟机的guest OS运行至某个Errata的Errata处理逻辑时,可根据CPU类型收集单元1011收集到的CPU类型信息确定当前异构CPU架构存在的所有Errata,若异构CPU架构存在该Errata,执行Errata修复逻辑(对此Errata进行修复),若异构CPU架构不存在该Errata,执行非修复逻辑(即对此Errata不进行修复)。若将动态Errata管理开关置于开启状态,虚拟机的guest OS运行至某个Errata的Errata处理逻辑时,通过实时获取VCPU线程所在的物理CPU类型信息,基于VCPU线程所在的物理CPU类型信息确定是执行Errata修复逻辑,还是非修复逻辑。例如,基于VCPU线程所在的物理CPU类型信息确定该类型的物理CPU存在的所有Errata,若该类型的物理CPU存在该Errata,执行Errata修复逻辑(对此Errata进行修复),若该类型的物理CPU不存在该Errata,执行非修复逻辑(即对此Errata不进行修复)。宿主机的host OS还可通过获取VCPU线程是否处于Errata处理逻辑中,以确定是否允许VCPU线程的调度,例如确定是否允许VCPU从当前类型的物理CPU调度在另一类型的物理CPU。若确定VCPU线程处于Errata处理逻辑中,禁止VCPU线程进行跨类型的物理CPU调度。

[0118] 虚拟化Errata管理前端模块103还用于获取VCPU类型维护单元1013维护的虚拟机的VCPU线程实时所在的物理CPU类型信息,例如,虚拟化Errata管理前端模块103可通过共享内存获取VCPU类型维护单元1013维护的虚拟机的VCPU线程实时所在的物理CPU类型信息。Errata管理系统104还用于在进行Errata处理之前,基于虚拟化Errata管理前端模块103获取的VCPU线程当前所在的物理CPU类型信息,确定是执行Errata修复逻辑(对此Errata进行修复),还是执行非修复逻辑(即对此Errata不进行修复)。

[0119] 在一些实施例中,虚拟化Errata管理前端模块103还用于维护虚拟机的VCPU线程的Errata处理上下文,Errata处理上下文用于表征Errata管理系统104在对Errata进行处理(执行Errata修复逻辑或执行非修复逻辑)。VCPU调度管理单元1014可获取虚拟化Errata

管理前端模块103的Errata处理上下文,进而可确定虚拟机的VCPU处于Errata处理逻辑中,禁止VCPU线程进行跨类型的物理CPU调度。例如,VCPU调度管理单元1014可通过共享内存获取虚拟化Errata管理前端模块103维护的Errata处理上下文。

[0120] 上述方案通过实时维护更新VCPU线程所在物理CPU类型信息以及VCPU线程的Errata处理上下文,使得VCPU线程在处于Errata处理逻辑中时,禁止进行跨类型的CPU调度,Errata处理完成之后,恢复VCPU线程的CPU自由调度,实现对各个虚拟机进行全局CPU资源调度与管理,可充分利用异构CPU架构的CPU资源,充分发挥异构CPU架构的算力&功耗的平衡,灵活支撑虚拟机的运维,提升虚拟机上部署的应用体验,且根据VCPU线程实时所在的物理CPU类型信息进行执行Errata修复逻辑或执行非修复逻辑,可避免执行冗余Errata修复逻辑,提升了虚拟机性能。

[0121] 在一些实施例中,对于CPU Errata与Timer Errata而言,还可以根据Errata管理方式的不同,将CPU Errata与Timer Errata分为两种类型:静态管理类型Errata与动态管理类型Errata。即根据Errata管理方式的不同,某些CPU Errata与Timer Errata可被划分为静态管理类型Errata,某些CPU Errata与Timer Errata可被划分为动态管理类型Errata。在其他实施例中,也可以将所有的CPU Errata与Timer Errata划分为动态管理类型Errata,也可以将所有的CPU Errata与Timer Errata划分为静态管理类型Errata。

[0122] 对于静态管理类型Errata,可以进行统一的Errata修复,例如,虚拟机可以在gust OS启动时,对CPU架构中存在的各个静态管理类型Errata进行识别与修复,或者虚拟机的guest OS运行至某个静态管理类型Errata的Errata处理逻辑时,若识别到CPU架构中存在该Errata,执行Errata修复逻辑,若识别到CPU架构中不存在该Errata,执行非修复逻辑。

[0123] 对于动态管理类型Errata,虚拟机的guest OS运行至某个Errata的Errata处理逻辑时,需实时感知CPU架构的CPU类型信息,确定是否执行Errata修复逻辑,若不存在此Errata,执行非修复逻辑,避免执行冗余Errata修复逻辑,可提升虚拟机性能。

[0124] 例如,对于图5a所示的方案而言,对于静态管理类型Errata,虚拟机可以在gust OS启动时,对各个静态管理类型Errata进行统一的识别与修复,或者虚拟机的guest OS运行至某个Errata(静态管理类型Errata)的Errata处理逻辑时,若识别到CPU架构中存在该Errata,执行Errata修复逻辑,若识别到CPU架构中不存在该Errata,执行非修复逻辑。对于动态管理类型Errata,虚拟机的guest OS运行至某个Errata(动态管理类型Errata)的Errata处理逻辑时,可基于虚拟化Errata管理前端模块103的Errata识别结果(虚拟化Errata管理前端模块103可基于虚拟化Errata管理后端模块101收集到的CPU类型信息与Timer Errata信息识别CPU架构中存在的所有Errata),确定是执行Errata修复逻辑,或是执行非修复逻辑。例如,在虚拟机的guest OS运行至某个动态管理类型Errata的Errata处理逻辑时,若确定CPU架构中不存在该动态管理类型Errata,执行非修复逻辑;在虚拟机的guest OS运行至某个动态管理类型Errata的Errata处理逻辑时,若确定CPU架构中存在该动态管理类型Errata,执行Errata修复逻辑。

[0125] 例如,对于图5b所示的方案而言,对于静态管理类型Errata,虚拟机可以在gust OS启动时,对各个静态管理类型Errata进行统一的识别与修复,或者虚拟机的guest OS运行至某个Errata的Errata处理逻辑时,若识别到CPU架构中存在该Errata,执行Errata修复逻辑,若识别到CPU架构中不存在该Errata,执行非修复逻辑。对于动态管理类型Errata,虚拟

机的guest OS运行至某个Errata的Errata处理逻辑时,可基于VCPU线程当前所在的物理CPU类型确定是执行Errata修复逻辑,或是执行非修复逻辑。例如,在虚拟机的guest OS运行至某个动态管理类型Errata的Errata处理逻辑时,通过获取VCPU线程当前所在的物理CPU类型,进而可基于物理CPU类型确定该类型的物理CPU存在的所有Errata,若确定该类型的物理CPU中不存在该动态管理类型Errata,执行非修复逻辑;在虚拟机的guest OS运行至某个动态管理类型Errata的Errata处理逻辑时,若确定该类型的物理CPU中存在该动态管理类型Errata,执行Errata修复逻辑。

[0126] 下面结合图6,示例性的介绍本申请一实施例提供的电子设备中的虚拟机与VMM的交互示意图。

[0127] 在该实施例中,电子设备10的软件层可部署有VMM及至少一个虚拟机。动态Errata管理开关处于关闭状态,电子设备10的CPU架构可以是异构CPU架构,也可以是同构CPU架构,本申请对此不作限定。虚拟机与VMM之间进行交互的过程可以包括:

[0128] 61,VMM启动虚拟化Errata管理后端模块101。

[0129] 在一些实施例中,虚拟化Errata管理后端模块101可基于内核的虚拟机(kernel-based virtual machine,KVM)进行初始化与启动。

[0130] 62,虚拟化Errata管理后端模块101收集CPU类型信息与Timer Errata信息。

[0131] 在一些实施例中,CPU类型信息即CPU架构所包含的所有CPU的类型。虚拟化Errata管理后端模块101可通过遍历电子设备10中的所有CPU,读取每个CPU的系统寄存器来实现收集CPU类型信息。虚拟化Errata管理后端模块101还可通过解析CPU架构的固件的ACPI表或DT表,从ACPI表或DT表中读取Timer Errata相关信息字段,基于Timer Errata相关信息字段识别CPU存在的Timer Errata。虚拟化Errata管理后端模块101还可根据CPU类型信息识别CPU存在的Timer Errata。

[0132] 举例而言,host OS及gust OS中可包含有CPU errata表,CPU errata表记录了所有类型的CPU存在的CPU Errata,在host OS或gust OS启动之后,可基于收集的CPU类型信息与CPU errata表进行比对,确定收集的CPU类型存在的所有CPU Errata。host OS及gust OS的CPU Timer驱动中可能包含有Timer Errata表,可基于收集的CPU类型信息与Timer errata表进行比对,确定收集的CPU类型存在的所有Timer Errata。宿主机的CPU架构使用的固件的ACPI表或DT表中同样可能会记录有Timer Errata信息,通过从ACPI表或DT表中读取Timer Errata相关信息字段,基于Timer Errata相关信息字段识别存在的Timer Errata,通过上述两种方式来识别Timer Errata,可实现识别出CPU存在的所有Timer Errata。

[0133] 63,虚拟化Errata管理后端模块101将收集到的CPU类型信息与Timer Errata信息写入至共享存储区。

[0134] 在一些实施例中,共享存储区可以是指虚拟机与宿主机均可访问的存储区,虚拟化Errata管理后端模块101与虚拟化Errata管理前端模块103均可写入信息至共享存储区,或者从共享存储区读取信息。例如,共享存储区可以设置在电子设备10的内存或磁盘上。

[0135] 64,虚拟机启动虚拟化Errata管理前端模块103及Errata管理系统104。

[0136] 在一些实施例中,虚拟化Errata管理前端模块103可基于KVM进行初始化与启动。

[0137] 在一些实施例中,Errata管理系统104也可以随着虚拟机的gust OS一并启动。

[0138] 在图6中,交互步骤64与交互步骤61可以并列执行,也可以具有先后执行关系,本申请实施例对此不作限定。

[0139] 65,虚拟化Errata管理前端模块103从共享存储区读取CPU类型信息与Timer Errata信息。

[0140] 66,虚拟化Errata管理前端模块103根据CPU类型信息和Timer Errata信息确定CPU架构存在的Errata(CPU Errata与Timer Errata)。

[0141] 67,虚拟机的guest OS在运行至第一Errata的Errata处理逻辑时,Errata管理系统104基于虚拟化Errata管理前端模块103的Errata识别结果,对第一Errata进行处理。

[0142] 在一些实施例中,对第一Errata进行处理可包括执行Errata修复逻辑,对第一Errata进行修复,或执行非修复逻辑,不对第一Errata修复逻辑。例如,若确定CPU架构存在第一Errata,执行Errata修复逻辑,对第一Errata进行修复,若确定CPU架构不存在第一Errata,执行非修复逻辑。

[0143] 下面结合图7,示例性的介绍本申请另一实施例提供的虚拟机与VMM的交互示意图。

[0144] 在该实施例中,电子设备10的软件层可部署有VMM及至少一个虚拟机。动态Errata管理开关处于开启状态,电子设备10的CPU架构可以是异构CPU架构,也可以是同构CPU架构,本申请对此不作限定。虚拟机与VMM之间进行交互的过程可以包括。

[0145] 70,VMM启动虚拟化Errata管理后端模块101。

[0146] 71,虚拟化Errata管理后端模块101收集CPU类型信息与Timer Errata信息,及维护虚拟机的VCPU线程当前所在的物理CPU类型信息。

[0147] 在一些实施例中,VCPU线程当前所在的物理CPU类型信息即VCPU线程当前所在的CPU的类型。虚拟化Errata管理后端模块101维护虚拟机的VCPU线程实时所在的物理CPU类型信息。当VCPU线程调度到其他CPU时,虚拟化Errata管理后端模块101可实时地进行VCPU线程所在的物理CPU类型信息的更新。

[0148] 72,虚拟化Errata管理后端模块101将收集到的CPU类型信息与Timer Errata信息,及VCPU线程当前所在的物理CPU类型信息写入至共享存储区。

[0149] 73,虚拟机启动虚拟化Errata管理前端模块103及Errata管理系统104。

[0150] 在图7中,交互步骤70与交互步骤73可以并列执行,也可以具有先后执行关系,本申请实施例对此不作限定。

[0151] 74,虚拟化Errata管理前端模块103从共享存储区读取CPU类型信息、Timer Errata信息及VCPU线程当前所在的物理CPU类型信息。

[0152] 75,虚拟化Errata管理前端模块103根据CPU类型信息和Timer Errata信息确定CPU架构存在的Errata(CPU Errata与Timer Errata)。

[0153] 76,虚拟机的guest OS在运行至第二Errata的Errata处理逻辑时,Errata管理系统104基于VCPU线程当前所在的物理CPU类型信息,对第二Errata进行处理。

[0154] 在一些实施例中,对第二Errata进行处理可包括执行Errata修复逻辑,对第二Errata进行修复,或执行非修复逻辑,不对第二Errata修复逻辑。例如,可以基于CPU类型信息和Timer Errata信息确定CPU架构存在的所有Errata,基于VCPU线程当前所在的物理CPU类型信息,进一步确定VCPU所在类型的CPU存在的所有Errata,若确定VCPU所在类型的CPU

存在第二Errata,执行Errata修复逻辑,对第二Errata进行修复,若确定VCPU所在类型的CPU不存在第二Errata,执行非修复逻辑。

[0155] 77,虚拟化Errata管理前端模块103将第二Errata处理上下文写入至共享存储区。

[0156] 在一些实施例中,第二Errata处理上下文表征Errata管理系统104在对第二Errata进行处理(执行Errata修复逻辑或执行非修复逻辑)。

[0157] 78,虚拟化Errata管理后端模块101从共享存储区读取第二Errata处理上下文。

[0158] 79,虚拟化Errata管理后端模块101基于第二Errata处理上下文管理虚拟机的VCPU线程的调度。

[0159] 在一些实施例中,虚拟化Errata管理后端模块101基于第二Errata处理上下文可获知前端在对第二Errata进行处理,禁止虚拟机的VCPU线程进行跨类型的CPU调度,即仅允许进行同类型的CPU调度。

[0160] 在一些实施例中,前端在对第二Errata进行处理的过程中,虚拟化Errata管理后端模块101还可以禁止VCPU线程进行CPU调度,即同类型的CPU调度亦不允许。例如,对于同构CPU架构,前端在对第二Errata进行处理的过程中,虚拟化Errata管理后端模块101禁止VCPU线程进行CPU调度。

[0161] 在一些实施例中,虚拟化Errata管理前端模块103在第二Errata处理完毕之后,同样可以将第二Errata处理完毕上下文写入至共享存储区。虚拟化Errata管理后端模块101可以基于第二Errata处理完毕上下文,允许VCPU线程进行调度(同类型的CPU调度、跨类型的CPU调度)。

[0162] 参照图8所示,本申请一实施例提供一种虚拟机处理勘误表的方法,应用于电子设备10,电子设备10包括由至少一种物理CPU构成的CPU架构,电子设备10运行有虚拟机及用于管理虚拟机的VMM。本实施例中,虚拟机处理勘误表的方法可以包括:

[0163] 步骤S81,VMM获取CPU架构的Errata相关信息,及将Errata相关信息共享给虚拟机。

[0164] 在一些实施例中,Errata相关信息可包括CPU类型信息、Timer Errata信息等。以CPU架构存在的Errata包括CPU Errata与Timer Errata为例,Errata相关信息可包括CPU类型信息与Timer Errata信息。

[0165] CPU类型信息包括CPU架构中的每种物理CPU的类型信息,VMM可通过遍历CPU架构中的每个物理CPU的系统寄存器,从中读取每种物理CPU的类型信息。Timer Errata信息包括第一Timer Errata信息与第二Timer Errata信息。VMM可将CPU类型信息与预设Timer Errata表进行比对,得到与CPU类型信息对应的第一Timer Errata信息。VMM还可获取CPU架构中的固件的ACPI表或DT表,从ACPI表或DT表中读取Timer Errata相关信息字段,及基于Timer Errata相关信息字段解析得到第二Timer Errata信息。

[0166] 例如,VMM包括虚拟化Errata管理后端模块101,虚拟机包括虚拟化Errata管理前端模块103,虚拟化Errata管理后端模块101遍历每个物理CPU的系统寄存器,从中读取每个物理CPU的类型信息,进而可汇总得到每种物理CPU的类型信息。虚拟化Errata管理后端模块101从CPU架构中的固件的ACPI表或DT表读取Timer Errata相关信息字段,对Timer Errata相关信息字段进行解析得到第二Timer Errata信息,虚拟化Errata管理后端模块101还可将CPU类型信息与预设Timer Errata表进行比对,得到第一Timer Errata信息。

[0167] 在一些实施例中,电子设备10上可配置有虚拟机及VMM均具有访问权限的共享存储区,例如共享内存。VMM(例如,VMM的虚拟化Errata管理后端模块101)可将Errata相关信息写入至共享存储区,虚拟机(例如,虚拟机的虚拟化Errata管理前端模块103)可从共享存储区读取VMM写入的Errata相关信息。

[0168] 步骤S82,虚拟机基于Errata相关信息识别CPU架构存在的Errata。

[0169] 在一些实施例中,以CPU架构存在的Errata包括CPU Errata及Timer Errata为例,但本申请实施例并不以此为限,随着技术的发展,还可能不存在其他类型的Errata,Errata相关信息还可包括用于识别其他类型的Errata的信息,本申请对此不作限定。

[0170] 虚拟机可将CPU类型信息与预设CPU Errata表进行比对,得到CPU架构存在的所有CPU Errata。虚拟机还可基于Timer Errata信息,得到CPU架构存在的所有Timer Errata。

[0171] 步骤S83,虚拟机对识别到的Errata进行修复处理。

[0172] 在一些实施例中,电子设备的host OS及虚拟机的guest OS可包含各种Errata的Errata处理逻辑,即各种Errata的Errata处理逻辑预先写入在host OS与guest OS中,这些Errata包含了CPU架构存在的Errata,以及CPU架构不存在的Errata。Errata按照类型划分可包括CPU Errata、Timer Errata等等,Errata还可按照管理方式进行划分,可划分为静态管理类型Errata与动态管理类型Errata。以第一Errata为静态管理类型Errata为例,在虚拟机的guest OS运行至第一Errata的Errata处理逻辑,且确定CPU架构存在第一Errata时,执行Errata修复逻辑。以第一Errata为动态管理类型Errata为例,虚拟机的guest OS运行至第一Errata的Errata处理逻辑时,基于CPU架构的CPU类型信息,确定是否执行第一Errata的Errata修复逻辑。例如,基于CPU类型信息确定CPU架构存在第一Errata,执行第一Errata的Errata修复逻辑,对第一Errata进行修复,若确定CPU架构不存在第一Errata,执行第一Errata的非修复逻辑,即不对第一Errata进行修复,避免执行冗余Errata修复逻辑,可提升虚拟机的性能。

[0173] 在一些实施例中,虚拟机还配置有动态Errata管理开关,通过动态Errata管理开关来选择是否开启Errata动态处理机制。例如,虚拟机可以响应于用户操作,将动态Errata管理开关置于关闭状态或者开启状态。若将动态Errata管理开关置于关闭状态,即关闭Errata动态处理机制,虚拟机的guest OS运行至某个动态管理类型Errata的Errata处理逻辑时,根据CPU类型信息确定是否执行Errata修复逻辑。例如,若基于CPU类型信息确定CPU架构存在该动态管理类型Errata,执行Errata修复逻辑,若基于CPU类型信息确定CPU架构不存在该动态管理类型Errata,执行非修复逻辑。

[0174] 在一些实施例中,若将动态Errata管理开关置于开启状态,即开启Errata动态处理机制。VMM可在后端实时维护虚拟机的VCPU线程所在的物理CPU类型信息,及将VCPU线程所在的物理CPU类型信息共享给虚拟机(例如,可通过共享内存将VCPU线程所在的物理CPU类型信息共享给虚拟机),虚拟机可通过实时获取VMM维护的VCPU线程所在的物理CPU类型信息,基于VCPU线程所在的物理CPU类型信息确定是否执行Errata修复逻辑,避免执行冗余Errata修复逻辑,可进一步提升虚拟机的性能。

[0175] 例如,对于静态管理类型Errata,即使将动态Errata管理开关置于开启状态,在虚拟机的guest OS运行至某个静态管理类型Errata的Errata处理逻辑,且确定CPU架构存在该静态管理类型Errata时,可直接执行Errata修复逻辑。对于动态管理类型Errata,若将动

态Errata管理开关置于开启状态,虚拟机的guest OS运行至某个动态管理类型Errata的Errata处理逻辑时,可基于VCPU线程当前所在的物理CPU类型确定是否执行Errata修复逻辑。以第二Errata为动态管理类型Errata为例,虚拟机的guest OS运行至第二Errata的Errata处理逻辑时,虚拟机通过获取后端维护的VCPU线程当前所在的物理CPU类型,基于物理CPU类型确定该类型的物理CPU是否存在第二Errata,若确定该类型的物理CPU中不存在第二Errata,执行第二Errata的非修复逻辑,若确定该类型的物理CPU中存在第二Errata,执行第二Errata的Errata修复逻辑,可避免执行冗余Errata修复逻辑,提升虚拟机的性能。

[0176] 在一些实施例中,虚拟机在执行某个Errata的Errata修复逻辑或非修复逻辑时,虚拟机还可维护Errata处理上下文,虚拟机可将Errata处理上下文共享给VMM(例如虚拟机可基于共享内存将Errata处理上下文共享给VMM),VMM基于Errata处理上下文管理虚拟机的VCPU线程的调度。例如,VMM可基于Errata处理上下文,禁止虚拟机的VCPU线程进行跨类型的CPU调度,或者VMM基于Errata处理上下文,禁止虚拟机的VCPU线程进行CPU调度,保证了虚拟机的VCPU线程正确处理Errata,避免虚拟机的VCPU线程在处理某个Errata的非修复逻辑时,被调度到其他包含此Errata的CPU,导致Errata处理逻辑错误,影响虚拟机的正常运行。

[0177] 在一些实施例中,在虚拟机的VCPU线程完成Errata处理之后,虚拟机可维护Errata处理完毕上下文,及将Errata处理完毕上下文共享给VMM,VMM可基于Errata处理完毕上下文,恢复虚拟机的VCPU线程的CPU调度权限。例如,从禁止虚拟机的VCPU线程进行跨类型的CPU调度恢复为允许虚拟机的VCPU线程进行跨类型的CPU调度,从禁止虚拟机的VCPU线程进行CPU调度恢复为允许虚拟机的VCPU线程进行CPU调度。

[0178] 本申请实施例提供的电子设备10,内部存储器121可用于存储指令,处理器110可用于调用内部存储器121中的指令,使得电子设备10执行上述相关方法步骤实现上述实施例中的虚拟机处理勘误表的方法。

[0179] 本申请实施例还提供一种计算机存储介质,所述计算机存储介质中存储有计算机指令,当所述计算机指令在电子设备10上运行时,使得电子设备10执行上述相关方法步骤实现上述实施例中的虚拟机处理勘误表的方法。

[0180] 本申请实施例还提供了一种计算机程序产品,当所述计算机程序产品在电子设备上运行时,使得电子设备执行上述相关步骤,以实现上述实施例中的虚拟机处理勘误表的方法。

[0181] 另外,本申请实施例还提供一种装置,这个装置具体可以是芯片,组件或模块,所述装置可包括相连的处理器和存储器;其中,存储器用于存储计算机执行指令,当装置运行时,处理器可执行存储器存储的计算机执行指令,以使芯片执行上述各方法实施例中的虚拟机处理勘误表的方法。

[0182] 其中,本申请实施例提供的计算机存储介质、计算机程序产品或芯片均用于执行上文所提供的对应的方法,因此,其所能达到的有益效果可参考上文所提供的对应的方法中的有益效果,此处不再赘述。

[0183] 通过以上的实施方式的描述,所属领域的技术人员可以清楚地了解到,为描述的方便和简洁,仅以上述各功能模块的划分进行举例说明,实际应用中,可以根据需要而将上述功能分配由不同的功能模块完成,即将装置的内部结构划分成不同的功能模块,以完成

以上描述的全部或者部分功能。本申请实施例所称的模块可以是指能够完成特定功能的一系列计算机程序指令段,也可以是计算机程序指令段与硬件配合形成的功能模块。

[0184] 在本申请所提供的几个实施例中,应所述理解到,所揭露的装置和方法,可以通过其它的方式实现。例如,以上所描述的装置实施例是示意性的,例如,所述模块或单元的划分,为一种逻辑功能划分,实际实现时可以有另外的划分方式,例如多个单元或组件可以结合或者可以集成到另一个装置,或一些特征可以忽略,或不执行。另一点,所显示或讨论的相互之间的耦合或直接耦合或通信连接可以是通过一些接口,装置或单元的间接耦合或通信连接,可以是电性,机械或其它的形式。

[0185] 所述作为分离部件说明的单元可以是或者也可以不是物理上分开的,作为单元显示的部件可以是一个物理单元或多个物理单元,即可以位于一个地方,或者也可以分布到多个不同地方。可以根据实际的需要选择其中的部分或者全部单元来实现本实施例方案的目的。

[0186] 另外,在本申请各个实施例中的各功能单元可以集成在一个处理单元中,也可以是各个单元单独物理存在,也可以两个或两个以上单元集成在一个单元中。上述集成的单元既可以采用硬件的形式实现,也可以采用软件功能单元的形式实现。

[0187] 所述集成的单元如果以软件功能单元的形式实现并作为独立的产品销售或使用,可以存储在一个可读取存储介质中。基于这样的理解,本申请实施例的技术方案本质上或者说对现有技术做出贡献的部分或者所述技术方案的全部或部分可以以软件产品的形式体现出来,所述软件产品存储在一个存储介质中,包括若干指令用以使得一个设备(可以是单片机,芯片等)或处理器(processor)执行本申请各个实施例所述方法的全部或部分步骤。而前述的存储介质包括:U盘、移动硬盘、只读存储器(Read-Only Memory,ROM)、随机存取存储器(Random Access Memory,RAM)、磁碟或者光盘等各种可以存储程序代码的介质。

[0188] 以上所述,仅为本申请的具体实施方式,但本申请的保护范围并不局限于此,任何在本申请揭露的技术范围内的变化或替换,都应涵盖在本申请的保护范围之内。

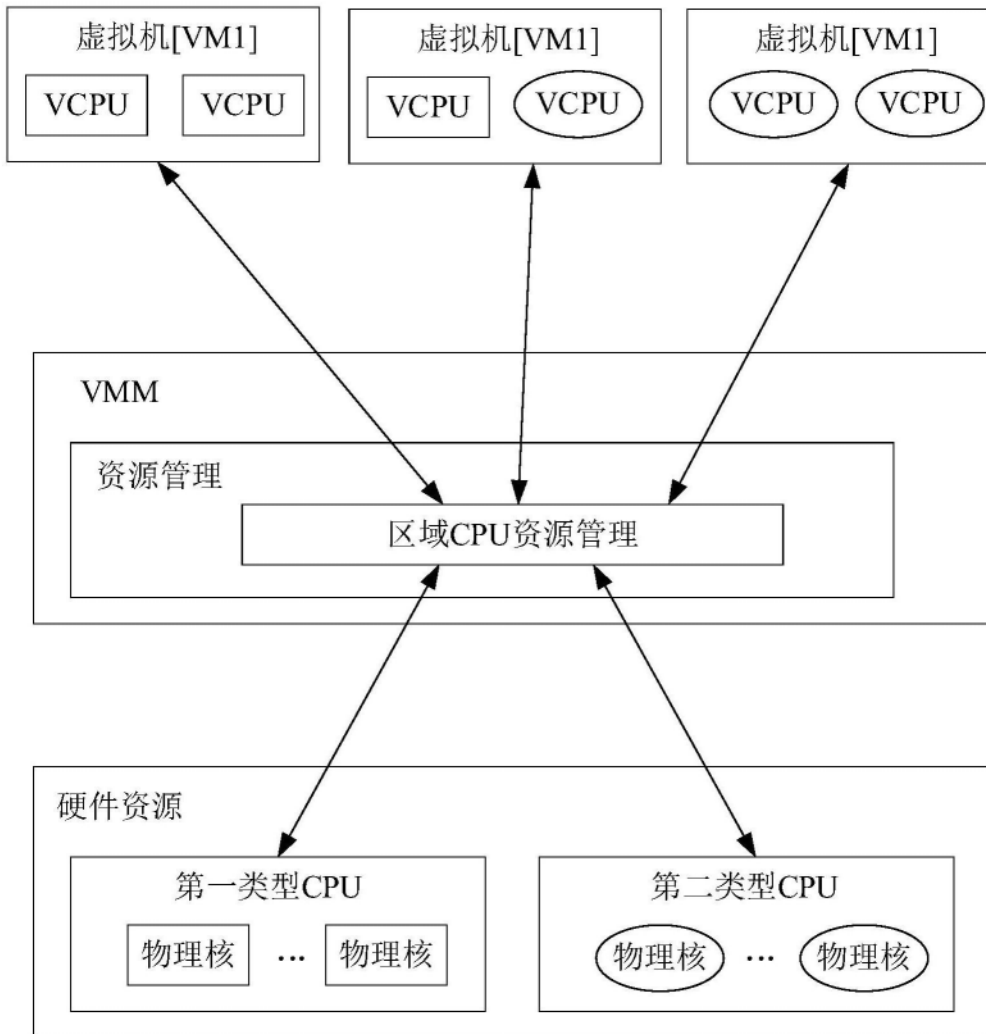


图1

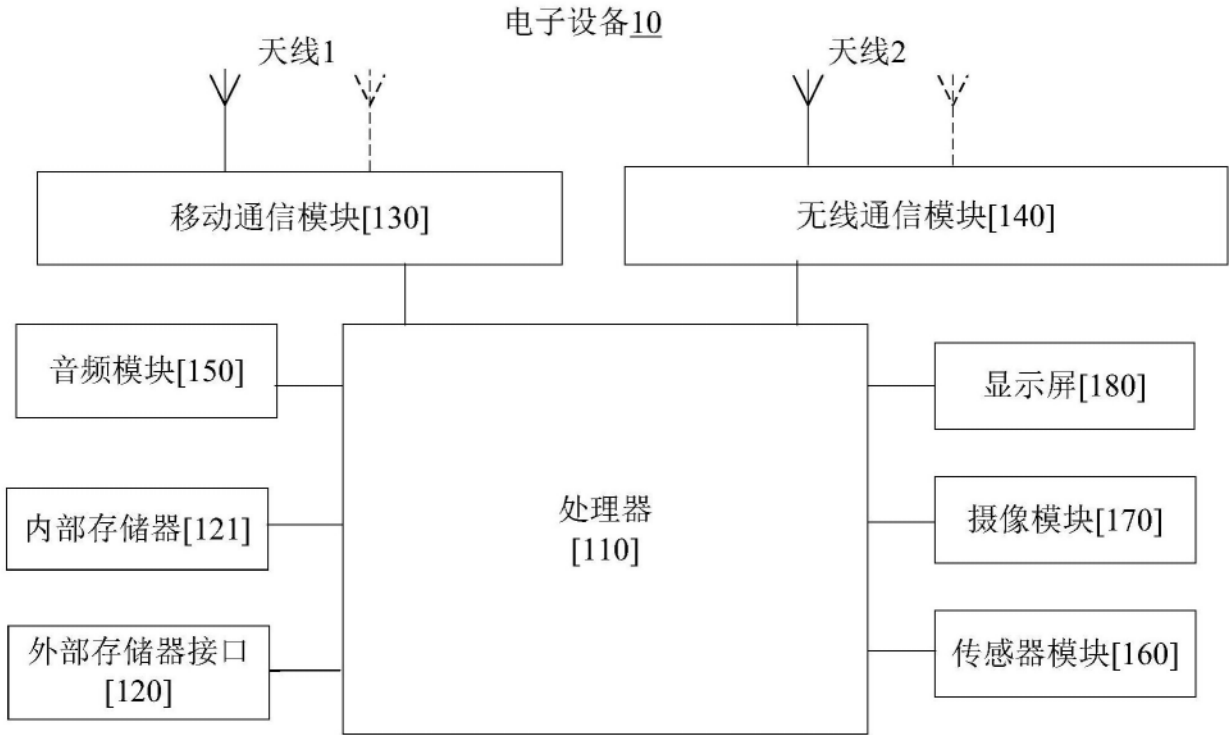


图2

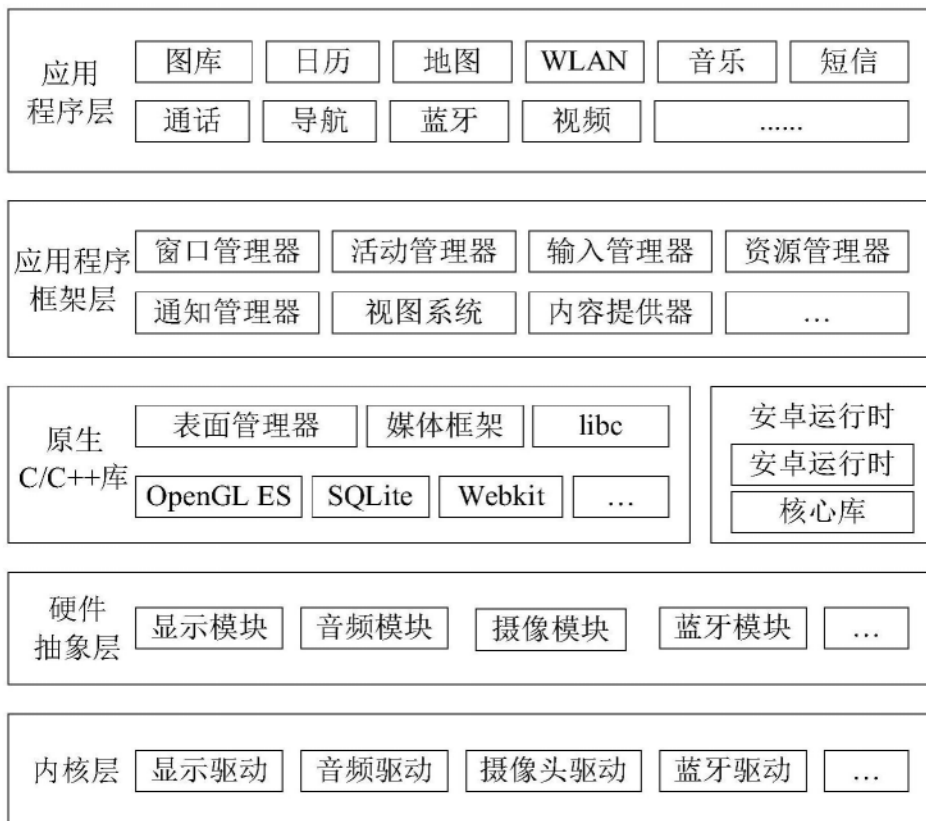


图3

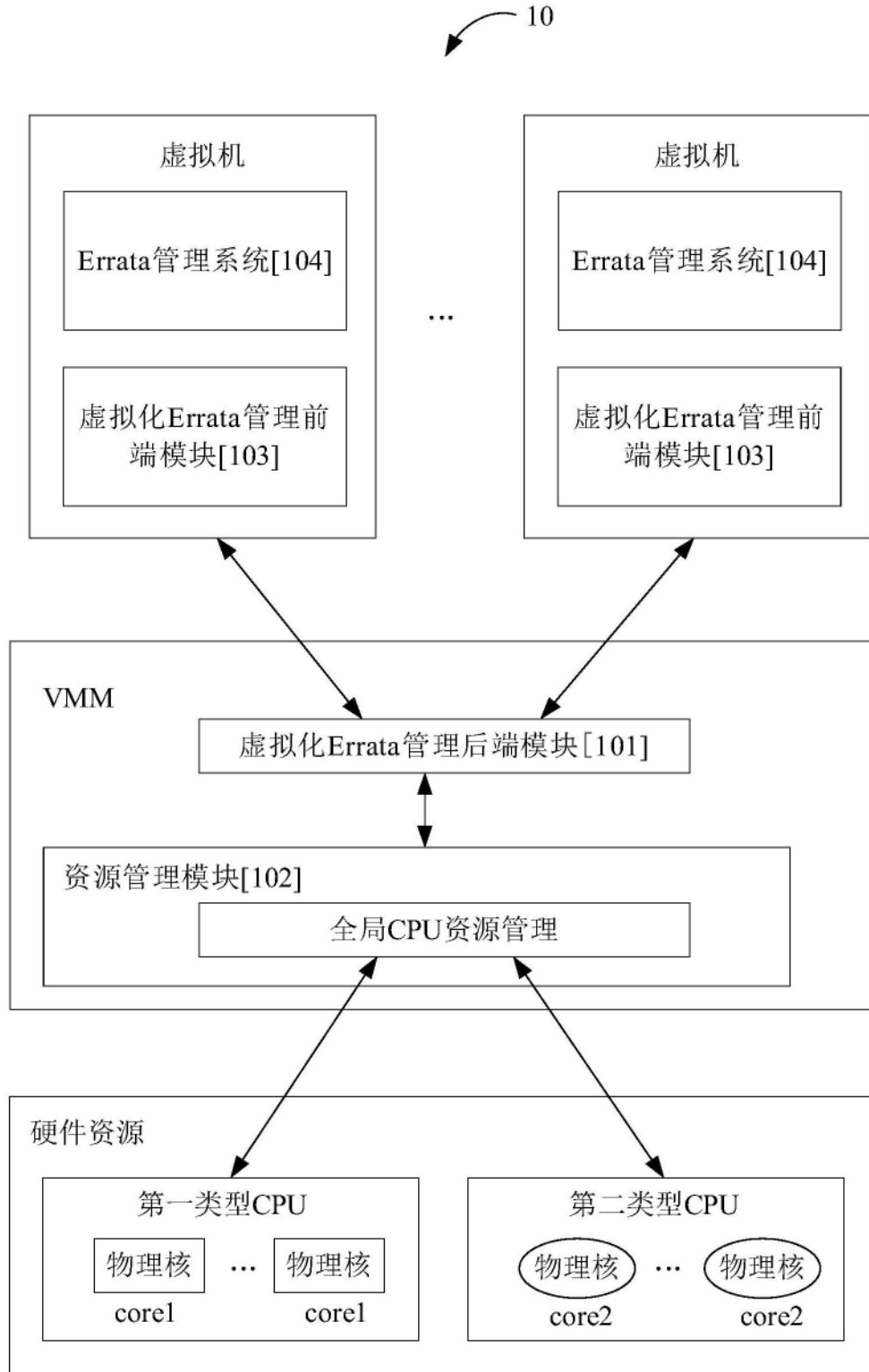


图4

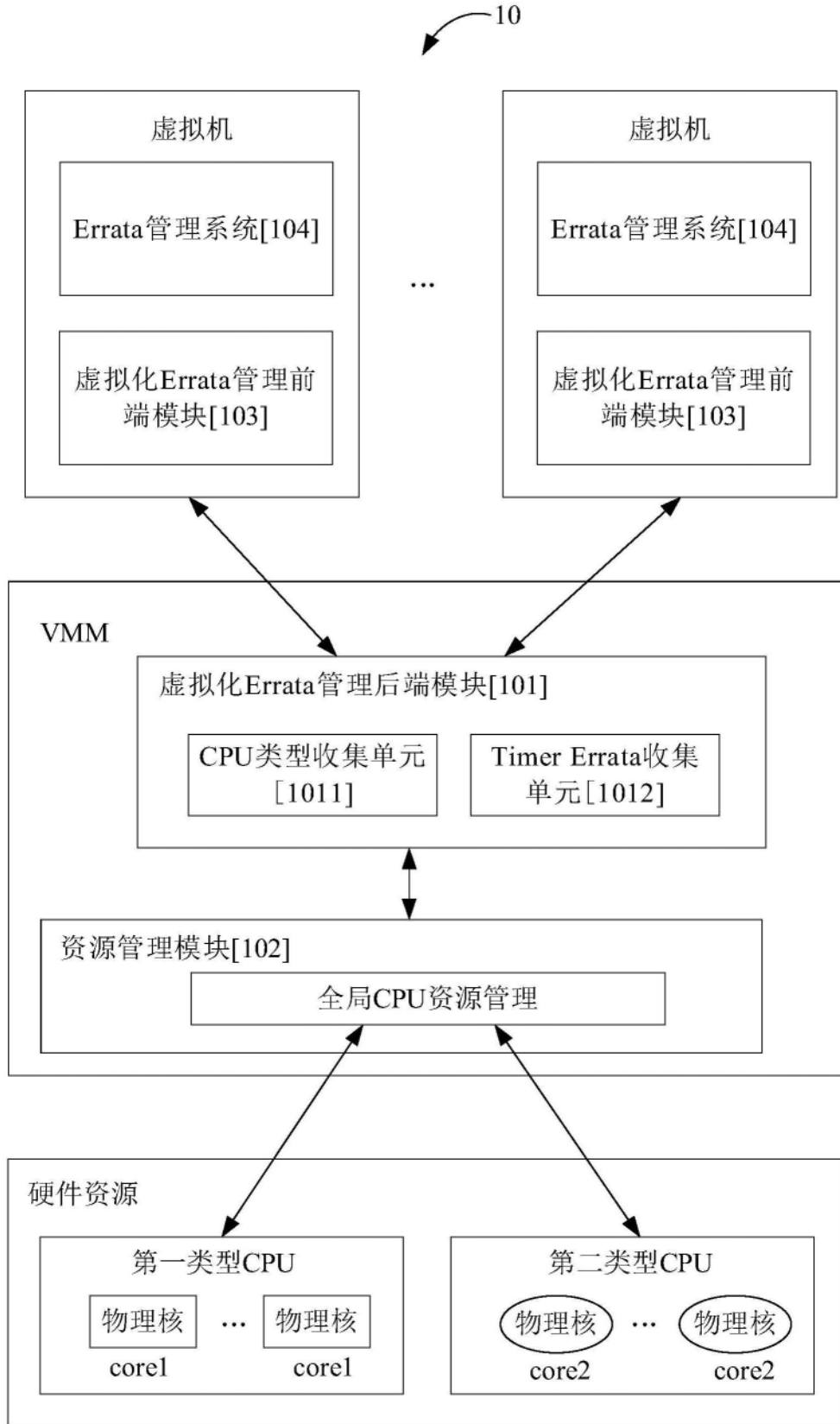


图5a

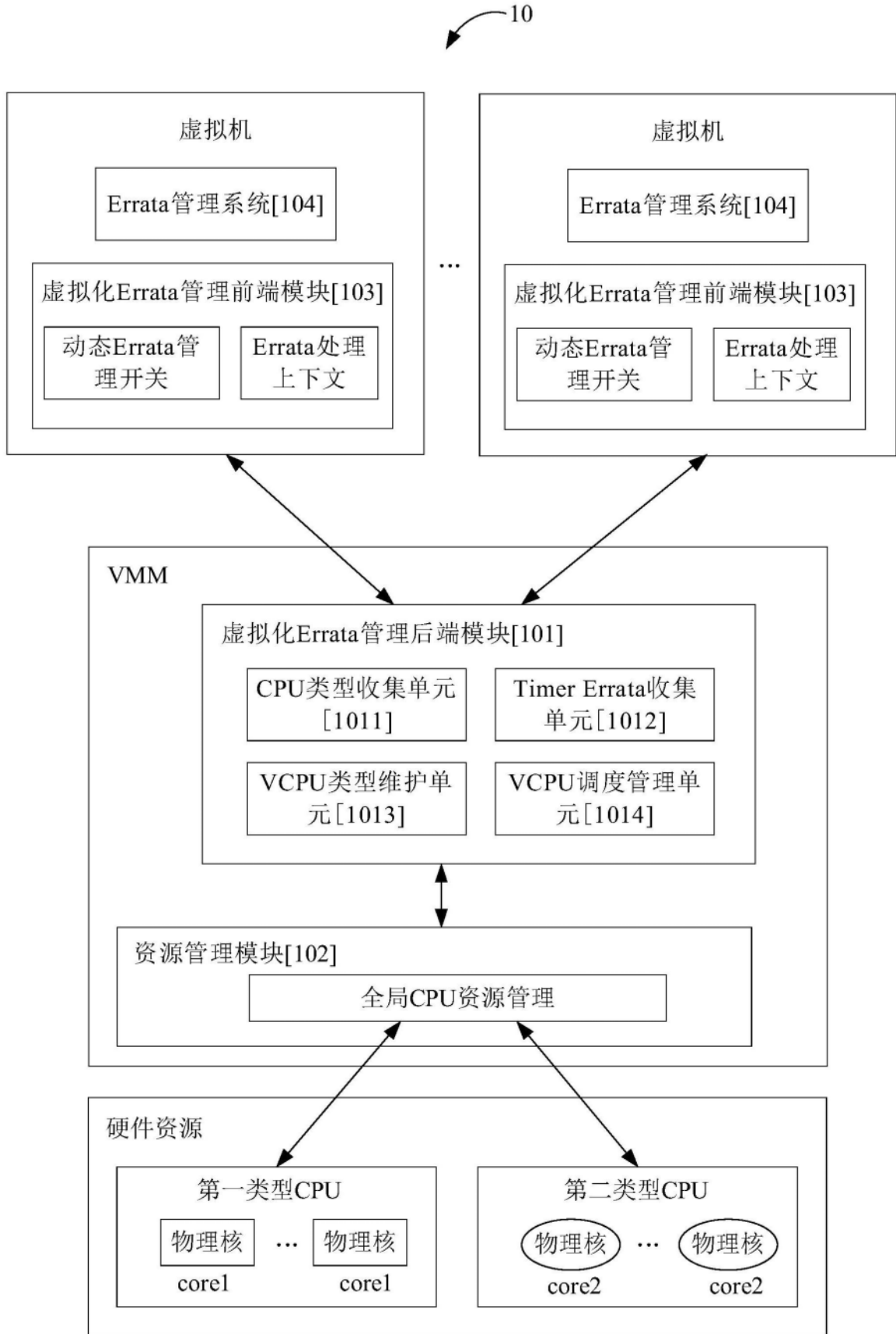


图5b

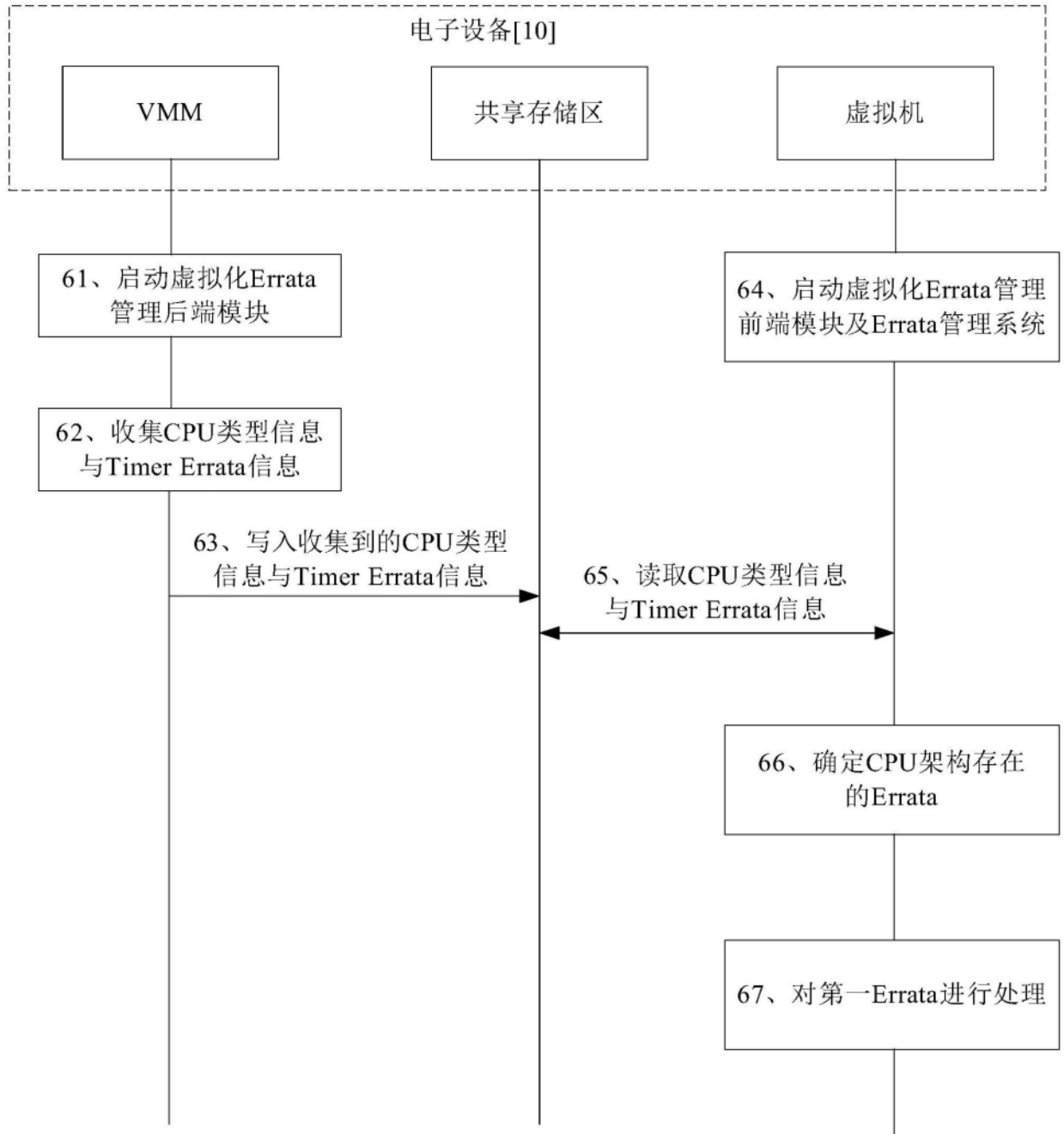


图6

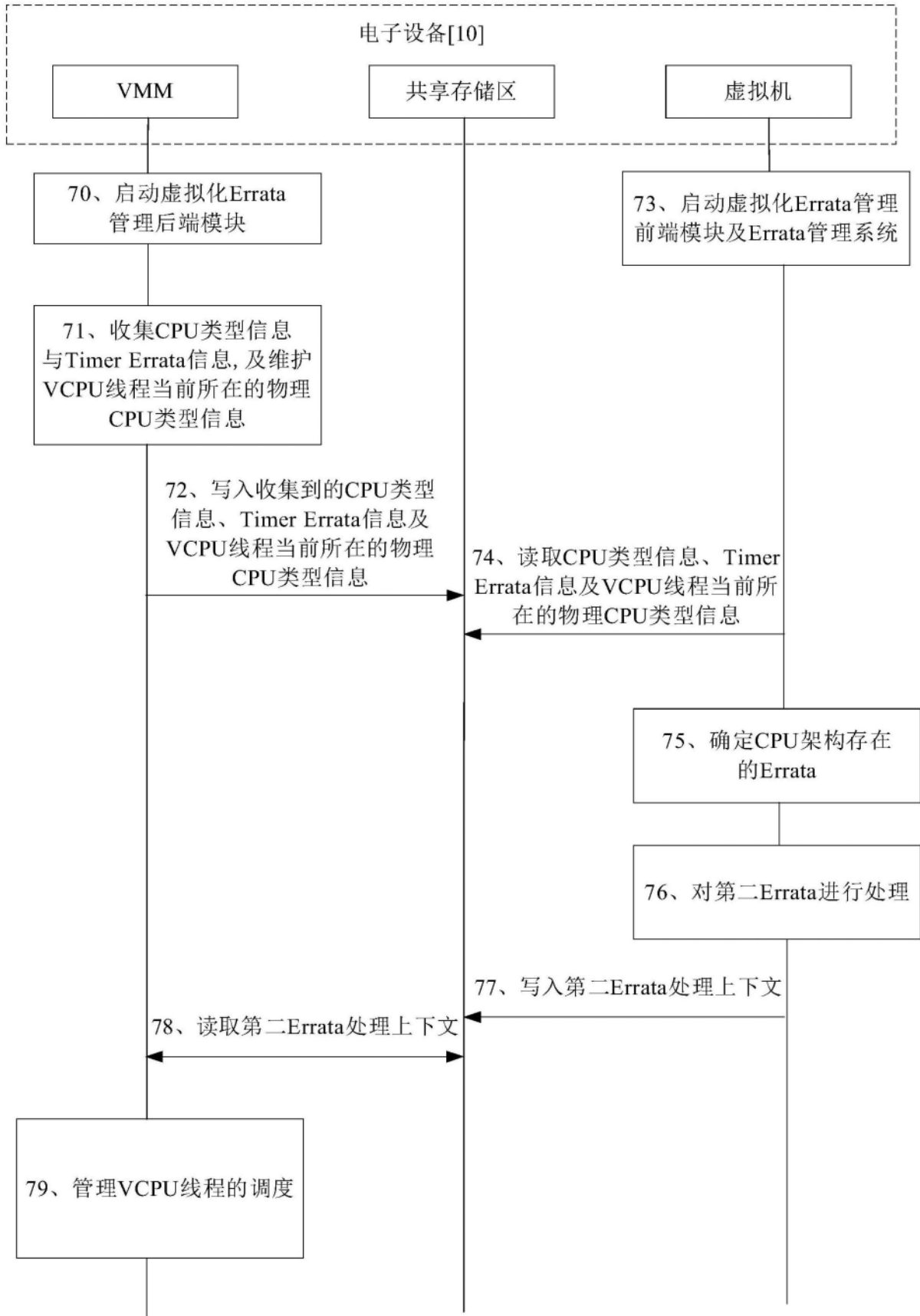


图7

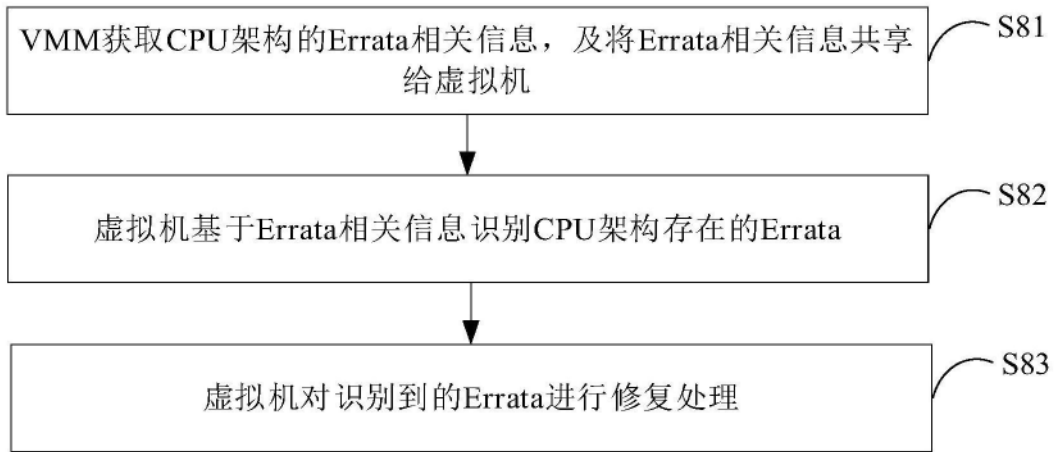


图8