



(12)发明专利申请

(10)申请公布号 CN 110162377 A

(43)申请公布日 2019.08.23

(21)申请号 201810152182.2

(22)申请日 2018.02.14

(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 刘晓建

(74)专利代理机构 深圳市深佳知识产权代理事务所(普通合伙) 44285

代理人 王仲凯

(51) Int. Cl.

G06F 9/455(2006.01)

G06F 9/48(2006.01)

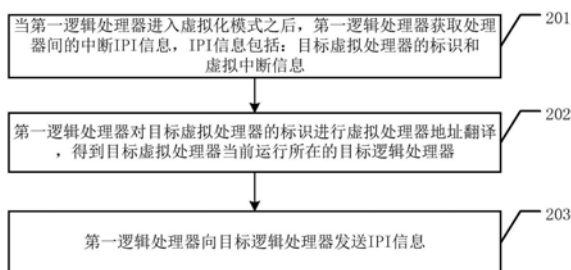
权利要求书3页 说明书24页 附图7页

(54)发明名称

一种通信方法和逻辑处理器

(57)摘要

本申请实施例公开了一种通信方法和逻辑处理器,用于减少对发送方的处理器资源占用和IPI的最终传输延迟。本申请实施例提供一种通信方法,包括:当第一逻辑处理器进入虚拟化模式之后,所述第一逻辑处理器获取处理器间中断IPI信息,所述IPI信息包括:目标虚拟处理器的标识和虚拟中断信息;所述第一逻辑处理器对所述目标虚拟处理器的标识进行虚拟处理器地址翻译,得到所述目标虚拟处理器当前运行所在的目标逻辑处理器;所述第一逻辑处理器向所述目标逻辑处理器发送所述IPI信息。



1. 一种通信方法,其特征在于,包括:

当第一逻辑处理器进入虚拟化模式之后,所述第一逻辑处理器获取处理器间中断IPI信息,所述IPI信息包括:目标虚拟处理器的标识和虚拟中断信息;

所述第一逻辑处理器对所述目标虚拟处理器的标识进行虚拟处理器地址翻译,得到所述目标虚拟处理器当前运行所在的目标逻辑处理器;

所述第一逻辑处理器向所述目标逻辑处理器发送所述IPI信息。

2. 根据权利要求1所述的方法,其特征在于,所述方法还包括:

所述第一逻辑处理器对所述目标虚拟处理器的标识进行虚拟处理器地址翻译之后,确定所述目标虚拟处理器没有在任何一个逻辑处理器运行时,所述第一逻辑处理器存储所述IPI信息,并退出所述虚拟化模式;

所述第一逻辑处理器通知虚拟机监控器VMM所述第一逻辑处理器存储了所述IPI信息。

3. 根据权利要求1所述的方法,其特征在于,所述方法还包括:

当所述第一逻辑处理器向所述目标逻辑处理器发送所述IPI信息失败时,所述第一逻辑处理器存储所述IPI信息,并退出所述虚拟化模式;

所述第一逻辑处理器通知VMM所述第一逻辑处理器存储了所述IPI信息。

4. 根据权利要求1所述的方法,其特征在于,所述方法还包括:

所述第一逻辑处理器向所述目标逻辑处理器发送所述目标虚拟处理器所在的虚拟机VM的标识。

5. 根据权利要求1至4中任一项所述的方法,其特征在于,所述第一逻辑处理器对所述目标虚拟处理器的标识进行虚拟处理器地址翻译,包括:

所述第一逻辑处理器通过查询虚拟处理器到逻辑处理器的路由表,对所述目标虚拟处理器的标识进行虚拟处理器地址翻译。

6. 根据权利要求5所述的方法,其特征在于,所述虚拟处理器到逻辑处理器的路由表存储在所述第一逻辑处理器的存储区域中;或,

所述虚拟处理器到逻辑处理器的路由表存储在所述第一逻辑处理器所在的可用于物理插拔的处理器存储区域中;或,

所述虚拟处理器到逻辑处理器的路由表存储到内存中。

7. 根据权利要求5所述的方法,其特征在于,所述第一逻辑处理器具有用于修改所述虚拟处理器到逻辑处理器的路由表的接口。

8. 一种通信方法,其特征在于,包括:

当第二逻辑处理器进入虚拟化模式之后,所述第二逻辑处理器接收第一逻辑处理器发送的处理器中断IPI信息,所述IPI信息包括:目标虚拟处理器的标识和虚拟中断信息;

所述第二逻辑处理器根据所述虚拟中断信息对第二虚拟处理器进行虚拟中断注入处理,所述第二虚拟处理器是所述第二逻辑处理器进入虚拟模式后正在运行的虚拟处理器。

9. 根据权利要求8所述的方法,其特征在于,所述方法还包括:

当所述第二虚拟处理器的标识与所述目标虚拟处理器的标识不相同时,所述第二逻辑处理器存储所述IPI信息,并退出所述虚拟化模式。

10. 根据权利要求8或9所述的方法,其特征在于,所述方法还包括:

所述第二逻辑处理器接收第一逻辑处理器发送的IPI信息之后,所述第二逻辑处理器

确定第二虚拟处理器的标识是否与所述目标虚拟处理器的标识相同；

当所述第二虚拟处理器的标识与所述目标虚拟处理器的标识相同时，触发执行如下步骤：所述第二逻辑处理器根据所述虚拟中断信息对第二虚拟处理器进行虚拟中断注入处理。

11. 一种逻辑处理器，其特征在于，所述逻辑处理器为第一逻辑处理器，所述第一逻辑处理器包括：

处理模块，用于当第一逻辑处理器进入虚拟化模式之后，获取处理器中断IPI信息，所述IPI信息包括：目标虚拟处理器的标识和虚拟中断信息；

所述处理模块，还用于对所述目标虚拟处理器的标识进行虚拟处理器地址翻译，得到所述目标虚拟处理器当前运行所在的目标逻辑处理器；

发送模块，用于向所述目标逻辑处理器发送所述IPI信息。

12. 根据权利要求11所述的逻辑处理器，其特征在于，所述处理模块，还用于对所述目标虚拟处理器的标识进行虚拟处理器地址翻译之后，确定所述目标虚拟处理器没有在任何逻辑处理器运行时，存储所述IPI信息，并退出所述虚拟化模式；

所述发送模块，还用于通知虚拟机监控器VMM所述第一逻辑处理器存储了所述IPI信息。

13. 根据权利要求11所述的逻辑处理器，其特征在于，所述处理模块，还用于当所述第一逻辑处理器向所述目标逻辑处理器发送所述IPI信息失败时，存储所述IPI信息，并退出所述虚拟化模式；

所述发送模块，还用于通知VMM所述第一逻辑处理器存储了所述IPI信息。

14. 根据权利要求11所述的逻辑处理器，其特征在于，所述发送模块，还用于向所述目标逻辑处理器发送所述目标虚拟处理器所在的虚拟机VM的标识。

15. 根据权利要求11至14中任一项所述的逻辑处理器，其特征在于，所述处理模块，具体用于通过查询虚拟处理器到逻辑处理器的路由表，对所述目标虚拟处理器的标识进行虚拟处理器地址翻译。

16. 根据权利要求15所述的逻辑处理器，其特征在于，所述虚拟处理器到逻辑处理器的路由表存储在所述第一逻辑处理器的存储区域中；或，

所述虚拟处理器到逻辑处理器的路由表存储在所述第一逻辑处理器所在的可用于物理插拔的处理器存储区域中；或，

所述虚拟处理器到逻辑处理器的路由表存储到内存中。

17. 根据权利要求15所述的逻辑处理器，其特征在于，所述第一逻辑处理器具有用于修改所述虚拟处理器到逻辑处理器的路由表的接口。

18. 一种逻辑处理器，其特征在于，所述逻辑处理器为第二逻辑处理器，所述第二逻辑处理器包括：

接收模块，用于当第二逻辑处理器进入虚拟化模式之后，接收第一逻辑处理器发送的处理器中断IPI信息，所述IPI信息包括：目标虚拟处理器的标识和虚拟中断信息；

处理模块，用于根据所述虚拟中断信息对所述第二虚拟处理器进行虚拟中断注入处理，所述第二虚拟处理器是所述第二逻辑处理器进入虚拟模式后正在运行的虚拟处理器。

19. 根据权利要求18所述的逻辑处理器，其特征在于，所述处理模块，还用于当所述第

二虚拟处理器的标识与所述目标虚拟处理器的标识不相同，存储所述IPI信息，并退出所述虚拟化模式。

20. 根据权利要求18或19所述的逻辑处理器，其特征在于，所述处理模块，还用于所述接收模块接收第一逻辑处理器发送的IPI信息之后，确定第二虚拟处理器的标识是否与所述目标虚拟处理器的标识相同；

所述处理模块，还用于当所述第二虚拟处理器的标识与所述目标虚拟处理器的标识相同时，触发执行如下步骤：根据所述虚拟中断信息对第二虚拟处理器进行虚拟中断注入处理。

21. 一种逻辑处理器，其特征在于，所述逻辑处理器包括：处理模块，存储模块；所述处理模块、所述存储模块之间进行相互的通信；

所述存储模块用于存储指令；

所述处理模块用于执行所述存储模块中的所述指令，执行如权利要求1至7、或权利要求8至10中任一项所述的方法。

22. 一种计算机可读存储介质，包括指令，当其在计算机上运行时，使得计算机执行如权利要求1至7、或权利要求8至10中任一项所述的方法。

23. 一种包含指令的计算机程序产品，当其在计算机上运行时，使得计算机执行如权利要求1至7、或权利要求8至10任一项所述的方法。

## 一种通信方法和逻辑处理器

### 技术领域

[0001] 本申请涉及计算机技术领域,尤其涉及一种通信方法和逻辑处理器。

### 背景技术

[0002] 在当前的处理器(Central Processing Unit,CPU)中,一个物理CPU模块对外呈现为多个指令处理部件,每个指令处理部件可以独立地执行一个指令流。这些指令处理部件一般被称作逻辑处理器(Logical CPU)。在计算机系统运行过程中,逻辑处理器之间需要进行异步通信,比如:一个逻辑处理器需要通知其它逻辑处理器进行任务切换。又如,一个逻辑处理器需要通知其它逻辑处理器清除某些转换监视缓冲器(Translation Lookaside Buffer,TLB)表项。又如,一个逻辑处理器需要通知其它逻辑处理器进行初始化。逻辑处理器间的这种异步通信一般被称作处理器间中断(Inter Processor Interrupt,IPI),或者软件中断(Software Generated Interrupt,SGI)。

[0003] 计算机虚拟化是将单台物理计算机呈现为网络上多台计算机的技术,网络用户可以使用、但实际上并不存在的计算机被称作虚拟计算机,简称为虚拟机(Virtual Machine,VM)。对网络用户而言,虚拟机和真实的物理计算机一样,具有处理器、内存、网卡、磁盘等硬件设备。虚拟机所包含的处理器被称作虚拟处理器(Virtual CPU),当逻辑处理器运行虚拟处理器对应的实体时,就对外体现为虚拟处理器正在运行。将虚拟处理器在运行时向其他虚拟处理器发送的IPI称为虚拟IPI。

[0004] 为了让多个虚拟机安全地共处于单台物理计算机上,需对虚拟机的能力进行限制,但又不能妨碍虚拟机的正常工作,这一过程被称作计算机系统虚拟化,负责实现计算机系统虚拟化的软件被称作虚拟机监控器(Virtual Machine Monitor,VMM)。

[0005] 硬件虚拟机(Hardware Virtual Machine,HVM)是一种依赖于硬件CPU支持实现的虚拟化,它的特点是:不要求虚拟机操作系统对计算机系统虚拟化提供配合,虚拟机操作系统可以像管理物理计算机一样管理虚拟机,如发送IPI。硬件对于虚拟化的支持可体现为:提供一个特定的处理器运行模式,如虚拟化模式。硬件处理器保证“运行于此模式下的虚拟机操作系统查看到的处理器状态与非虚拟化情形相同,HVM的虚拟机操作系统进行的特权操作要么能够直接完成,要么会导致逻辑处理器退出当前运行模式,转到VMM的处理逻辑”。此时当VMM需要让某个虚拟处理器处于运行状态时,VMM就令逻辑处理器进入虚拟化模式,逻辑处理器执行此虚拟处理器在运行状态需要执行的代码,逻辑处理器在发送IPI时需要退出虚拟化模式,由VMM向逻辑处理器注册“退出虚拟化模式后应当执行的代码入口”。

[0006] 现有技术中计算机系统虚拟化通过VMM软件实现,VMM软件采取与非虚拟化情形下相同的方式,一个虚拟处理器向目标虚拟处理器发送IPI是基于逻辑处理器发送IPI来实现的,而IPI是导致逻辑处理器退出虚拟化模式的操作,因此作为发送方的逻辑处理器在退出虚拟化模式时进行模式切换需要一定的时间,这会增大发送方的处理器资源占用和IPI的最终传输延迟。

## 发明内容

[0007] 本申请实施例提供了一种通信方法和逻辑处理器,用于减少对发送方的处理器资源占用和IPI的最终传输延迟。

[0008] 为解决上述技术问题,本申请实施例提供以下技术方案:

[0009] 第一方面,本申请实施例提供一种通信方法,其特征在于,包括:当第一逻辑处理器进入虚拟化模式之后,所述第一逻辑处理器获取处理器间中断IPI信息,所述IPI信息包括:目标虚拟处理器的标识和虚拟中断信息;所述第一逻辑处理器对所述目标虚拟处理器的标识进行虚拟处理器地址翻译,得到所述目标虚拟处理器当前运行所在的目标逻辑处理器;所述第一逻辑处理器向所述目标逻辑处理器发送所述IPI信息。

[0010] 在本申请实施例中,当第一逻辑处理器进入虚拟化模式之后,第一逻辑处理器获取处理器中断IPI信息,IPI信息包括:目标虚拟处理器的标识和虚拟中断信息;第一逻辑处理器对目标虚拟处理器的标识进行虚拟处理器地址翻译,得到目标虚拟处理器当前运行所在的目标逻辑处理器;第一逻辑处理器向目标逻辑处理器发送IPI信息。本申请实施例中,第一逻辑处理器可以对目标虚拟处理器的标识进行虚拟处理器地址翻译,从而可以确定出目标虚拟处理器当前运行所在的目标逻辑处理器,因此第一逻辑处理器可以在虚拟化模式直接向正在运行的目标虚拟处理器发送IPI信息,使得目标虚拟处理器可以被注入IPI,不需要通过VMM来实现对IPI信息进行发送,第一逻辑处理器也不需要退出虚拟化模式,因此可以减少对发送方的处理器资源占用和IPI的最终传输延迟。

[0011] 在本申请第一方面的一种可能设计中,所述方法还包括:所述第一逻辑处理器对所述目标虚拟处理器的标识进行虚拟处理器地址翻译之后,确定所述目标虚拟处理器没有在任何逻辑处理器运行时,所述第一逻辑处理器存储所述IPI信息,并退出所述虚拟化模式;所述第一逻辑处理器通知虚拟机监控器VMM所述第一逻辑处理器存储了所述IPI信息。其中,第一逻辑处理器对目标虚拟处理器的标识进行虚拟处理器地址翻译之后,确定目标虚拟处理器没有在任何逻辑处理器运行时,则说明该第一逻辑处理器无法确定出目标虚拟处理器当前运行所在的目标逻辑处理器,此时可以将IPI信息存储下来,例如存储到发送方异常信息存储区域中,使得运行于非虚拟化模式的vmm软件可以查询到IPI发送详情信息。

[0012] 在本申请第一方面的一种可能设计中,所述方法还包括:当所述第一逻辑处理器向所述目标逻辑处理器发送所述IPI信息失败时,所述第一逻辑处理器存储所述IPI信息,并退出所述虚拟化模式;所述第一逻辑处理器通知VMM所述第一逻辑处理器存储了所述IPI信息。其中,第一逻辑处理器可以设置“投送中断”时所使用的中断号,在逻辑处理器进入虚拟化模式后,执行IPI信息。退出虚拟化模式之后,则判断退出原因,若原因是“虚拟化模式下投送虚拟中断”,则由VMM使用中断号继续执行IPI中断信息。

[0013] 在本申请第一方面的一种可能设计中,所述方法还包括:所述第一逻辑处理器向所述目标逻辑处理器发送所述目标虚拟处理器所在的虚拟机VM的标识。其中,第一逻辑处理器还需要向目标逻辑处理器发送目标虚拟处理器所在的VM的标识,以使得接收方可以确定出该目标虚拟处理器所在的VM。

[0014] 在本申请第一方面的一种可能设计中,所述第一逻辑处理器对所述目标虚拟处理

器的标识进行虚拟处理器地址翻译,包括:所述第一逻辑处理器通过查询虚拟处理器到逻辑处理器的路由表,对所述目标虚拟处理器的标识进行虚拟处理器地址翻译。其中,在每个逻辑处理器上运行虚拟处理器时,每个逻辑处理器都对应一个根据虚拟处理器编号进行索引的虚拟处理器到逻辑处理器的路由表,一个逻辑处理器可以访问自己对应的虚拟处理器到逻辑处理器的路由表,从而完成从虚拟处理器到相应的逻辑处理器的地址翻译。

[0015] 在本申请第一方面的一种可能设计中,所述虚拟处理器到逻辑处理器的路由表存储在所述第一逻辑处理器的存储区域中;或,所述虚拟处理器到逻辑处理器的路由表存储在所述第一逻辑处理器所在的可用于物理插拔的处理器存储区域中;或,所述虚拟处理器到逻辑处理器的路由表存储到内存中。

[0016] 在本申请第一方面的一种可能设计中,所述第一逻辑处理器具有用于修改所述虚拟处理器到逻辑处理器的路由表的接口。

[0017] 第二方面,本申请实施例提供一种通信方法,包括:当第二逻辑处理器进入虚拟化模式之后,所述第二逻辑处理器接收第一逻辑处理器发送的处理器中断IPI信息,所述IPI信息包括:目标虚拟处理器的标识和虚拟中断信息;所述第二逻辑处理器根据所述虚拟中断信息对第二虚拟处理器进行虚拟中断注入处理,所述第二虚拟处理器是所述第二逻辑处理器进入虚拟模式后正在运行的虚拟处理器。

[0018] 本申请的前述实施例中,第一逻辑处理器可以对目标虚拟处理器的标识进行虚拟处理器地址翻译,从而可以确定出目标虚拟处理器当前运行所在的目标逻辑处理器,因此第一逻辑处理器可以在虚拟化模式直接向正在运行的目标虚拟处理器发送IPI信息,第二逻辑处理器作为目标逻辑处理器,该第二逻辑处理器上运行有该目标虚拟处理器,因此第二逻辑处理器可以对目标虚拟处理器进行IPI注入,不需要通过VMM来实现对IPI信息进行发送,第一逻辑处理器也不需要退出虚拟化模式,因此可以减少对发送方的处理器资源占用和IPI的最终传输延迟。

[0019] 在本申请第二方面的一种可能设计中,所述方法还包括:当所述第二虚拟处理器的标识与所述目标虚拟处理器的标识不相同,所述第二逻辑处理器存储所述IPI信息,并退出所述虚拟化模式。其中,第二逻辑处理器确定当前正在运行的第二虚拟处理器并不是目标虚拟处理器时,第二逻辑处理器可以将IPI信息存储下来,例如存储到接收方异常信息存储区域中,使得运行于非虚拟化模式的vmm软件可以查询到IPI发送详情信息。

[0020] 在本申请第二方面的一种可能设计中,所述方法还包括:所述第二逻辑处理器接收第一逻辑处理器发送的IPI信息之后,所述第二逻辑处理器确定第二虚拟处理器的标识是否与所述目标虚拟处理器的标识相同;当所述第二虚拟处理器的标识与所述目标虚拟处理器的标识相同时,触发执行如下步骤:所述第二逻辑处理器根据所述虚拟中断信息对第二虚拟处理器进行虚拟中断注入处理。

[0021] 在本申请的第三方面中,逻辑处理器的组成模块还可以执行前述第一方面以及各种可能的实现方式中所描述的步骤,详见前述对第一方面以及各种可能的实现方式中的说明。

[0022] 第三方面,本申请实施例提供一种逻辑处理器,所述逻辑处理器为第一逻辑处理器,所述第一逻辑处理器包括:处理模块,用于当第一逻辑处理器进入虚拟化模式之后,获取处理器中断IPI信息,所述IPI信息包括:目标虚拟处理器的标识和虚拟中断信息;所述处

理模块,还用于对所述目标虚拟处理器的标识进行虚拟处理器地址翻译,得到所述目标虚拟处理器当前运行所在的目标逻辑处理器;发送模块,用于向所述目标逻辑处理器发送所述IPI信息。

[0023] 在本申请第三方面的一种可能设计中,所述处理模块,还用于对所述目标虚拟处理器的标识进行虚拟处理器地址翻译之后,确定所述目标虚拟处理器没有在任何一个逻辑处理器运行时,存储所述IPI信息,并退出所述虚拟化模式;所述发送模块,还用于通知虚拟机监控器VMM所述第一逻辑处理器存储了所述IPI信息。

[0024] 在本申请第三方面的一种可能设计中,所述处理模块,还用于当所述第一逻辑处理器向所述目标逻辑处理器发送所述IPI信息失败时,存储所述IPI信息,并退出所述虚拟化模式;所述发送模块,还用于通知VMM所述第一逻辑处理器存储了所述IPI信息。

[0025] 在本申请第三方面的一种可能设计中,所述发送模块,还用于向所述目标逻辑处理器发送所述目标虚拟处理器所在的虚拟机VM的标识。

[0026] 在本申请第三方面的一种可能设计中,所述处理模块,具体用于通过查询虚拟处理器到逻辑处理器的路由表,对所述目标虚拟处理器的标识进行虚拟处理器地址翻译。

[0027] 在本申请第三方面的一种可能设计中,所述虚拟处理器到逻辑处理器的路由表存储在所述第一逻辑处理器的存储区域中;或,所述虚拟处理器到逻辑处理器的路由表存储在所述第一逻辑处理器所在的可用于物理插拔的处理器存储区域中;或,所述虚拟处理器到逻辑处理器的路由表存储到内存中。

[0028] 在本申请第三方面的一种可能设计中,所述第一逻辑处理器具有用于修改所述虚拟处理器到逻辑处理器的路由表的接口。

[0029] 第四方面,本申请实施例提供一种逻辑处理器,所述逻辑处理器为第二逻辑处理器,所述第二逻辑处理器包括:接收模块,用于当第二逻辑处理器进入虚拟化模式之后,接收第一逻辑处理器发送的处理器中断IPI信息,所述IPI信息包括:目标虚拟处理器的标识和虚拟中断信息;处理模块,用于根据所述虚拟中断信息对所述第二虚拟处理器进行虚拟中断注入处理,所述第二虚拟处理器是所述第二逻辑处理器进入虚拟模式后正在运行的虚拟处理器。

[0030] 在本申请第四方面的一种可能设计中,所述处理模块,还用于当所述第二虚拟处理器的标识与所述目标虚拟处理器的标识不相同,存储所述IPI信息,并退出所述虚拟化模式。

[0031] 在本申请第四方面的一种可能设计中,所述处理模块,还用于所述接收模块接收第一逻辑处理器发送的IPI信息之后,确定第二虚拟处理器的标识是否与所述目标虚拟处理器的标识相同;

[0032] 所述处理模块,还用于当所述第二虚拟处理器的标识与所述目标虚拟处理器的标识相同时,触发执行如下步骤:根据所述虚拟中断信息对第二虚拟处理器进行虚拟中断注入处理。

[0033] 在本申请的第四方面中,逻辑处理器的组成模块还可以执行前述第二方面以及各种可能的实现方式中所描述的步骤,详见前述对第二方面以及各种可能的实现方式中的说明。

[0034] 第五方面,本申请实施例提供了一种计算机可读存储介质,所述计算机可读存储



介质中存储有指令,当其在计算机上运行时,使得计算机执行上述各方面所述的方法。

[0035] 第六方面,本申请实施例提供了一种包含指令的计算机程序产品,当其在计算机上运行时,使得计算机执行上述各方面所述的方法。

[0036] 第七方面,本申请实施例提供一种通信装置,该通信装置可以包括终端设备或者芯片等实体,所述通信装置包括:处理器、存储器;所述存储器用于存储指令;所述处理器用于执行所述存储器中的所述指令,具体执行如前述第一方面或第二方面中任一项所述的方法。

[0037] 第八方面,本申请提供了一种芯片系统,该芯片系统包括处理器,用于支持网络设备实现上述方面中所涉及的功能,例如,例如发送或处理上述方法中所涉及的数据和/或信息。在一种可能的设计中,所述芯片系统还包括存储器,所述存储器,用于保存网络设备必要的程序指令和数据。该芯片系统,可以由芯片构成,也可以包括芯片和其他分立器件。

## 附图说明

[0038] 图1为本申请实施例提供的处理器的系统架构示意图;

[0039] 图2为本申请实施例提供的一种通信方法的方框流程示意图;

[0040] 图3为本申请实施例提供的另一种通信方法的方框流程示意图;

[0041] 图4为本申请实施例提供的逻辑处理器的基本架构示意图;

[0042] 图5为本申请实施例的发送方的逻辑处理器和接收方的逻辑处理器之间的一种交互流程示意图;

[0043] 图6为本申请实施例中发送方的逻辑处理器和接收方的逻辑处理器之间的另一种交互流程示意图;

[0044] 图7为本申请实施例中处理器socket包括的模块示意图;

[0045] 图8为本申请实施例提供的虚拟化模式下逻辑处理器执行IPI信息时的流程示意图;

[0046] 图9为本申请实施例提供的虚拟化模式下逻辑处理器检测到notify\_if中有信息后的处理流程示意图;

[0047] 图10为本申请实施例提供的虚拟化模式下逻辑处理器对于notify\_if请求的处理流程示意图;

[0048] 图11为本申请实施例提供的一种第一逻辑处理器的组成结构示意图;

[0049] 图12为本申请实施例提供的一种第二逻辑处理器的组成结构示意图。

## 具体实施方式

[0050] 本申请实施例提供了一种通信方法和逻辑处理器,用于减少对发送方的处理器资源占用和IPI的最终传输延迟。

[0051] 下面结合附图,对本申请的实施例进行描述。

[0052] 本申请的说明书和权利要求书及上述附图中的术语“第一”、“第二”等是用于区别类似的对象,而不必用于描述特定的顺序或先后次序。应该理解这样使用的术语在适当情况下可以互换,这仅仅是描述本申请的实施例中具有相同属性的对象在描述时所采用的区分方式。此外,术语“包括”和“具有”以及他们的任何变形,意图在于覆盖不排他的包含,以便

包含一系列单元的过程、方法、系统、产品或设备不必限于那些单元,而是可包括没有清楚地列出的或对于这些过程、方法、产品或设备固有的其它单元。

[0053] 首先对本申请实施例中的重要术语作出如下说明:

[0054] 虚拟机:在计算机科学中的体系结构里,是指一种特殊的软件,它可以在计算机平台和终端用户之间创建一种环境,而终端用户则是基于这个软件所创建的环境来操作软件。在计算机科学中,虚拟机是指可以像真实机器一样运行程序的计算机的软件实现。指通过软件模拟的具有完整硬件系统功能的、运行在一个独立环境中的完整计算机系统。一台物理机可以根据应用需要虚拟出多台虚拟机,实现在一台物理机上同时运行多个操作系统。对于每个操作系统,用户都可以进行虚拟的分区、配置。同时,用户可在多个操作系统间切换。虚拟机是一个软件计算机,类似于一台物理机,运行操作系统和应用。多个虚拟机能够在同一主机系统上同步运行。

[0055] 虚拟机监控器(Virtual Machine Monitor,VMM):VMM其实就是一个简单的操作系统。它提供很多抽象的虚拟机可以让多个操作系统同时运行。甚至这些操作系统也不需要相同。从系统的角度看,这个时候的每一个虚拟机其实相当于VMM这个OS(操作系统)的一个进程。

[0056] hyper-threading:超线程:采用另一个思路去提高CPU的性能,让CPU可以同时执行多重线程,就能够让CPU发挥更大效率,即所谓“超线程(Hyper-Threading,简称“HT”)”技术。超线程技术就是利用特殊的硬件指令,把两个逻辑内核模拟成两个物理芯片,让单个处理器都能使用线程级并行计算,进而兼容多线程操作系统和软件,减少了CPU的闲置时间,提高CPU的运行效率。采用超线程技术能同时执行两个线程,但它并不像两个真正的CPU那样,每个CPU都具有独立的资源。当两个线程都同时需要某一个资源时,其中一个要暂时停止,并让出资源,直到这些资源闲置后才能继续。因此超线程的性能并不等于两个CPU的性能。

[0057] Socket:插座。连接器的接收插座,用于插进插头。

[0058] 处理器socket:可以进行物理插拔的CPU硬件模块。

[0059] 处理器中断(Inter-Processor Interrupt,IPI):一个处理器向另外一个处理器发出的异步通知。在有的系统上,被称作SGI/软件中断。IPI是一种特别的中断。在对称多处理器(SMP,symmetric multiprocessing)环境下,它可以被任意一个处理器用来对另一个处理器产生中断。IPI典型地被用来实现高速缓存间的一致性同步(Cache Coherency Synchronization)和通知对方进行任务调度/抢先。

[0060] 对称多处理(symmetric multiprocessing,SMP):SMP指在一个计算机上汇集了一组处理器(多CPU),各CPU之间共享内存子系统以及总线结构。系统将任务队列对称地分布于多个CPU之上,所有的CPU都可以平等地访问内存、输入/输出(Input/Output,I/O)和外部设备。

[0061] 以下分别进行详细说明。

[0062] 请参阅图1所示,为本申请一个实施例提供的处理器的系统架构示意图,该系统架构可以包括:处理器和虚拟机,其中处理器是硬件,虚拟机是软件。处理器也称为中央处理单元,是计算机的运算和控制单元,它是解释和执行指令的部件,具有取址、译码和执行指令的基本功能,以及通过计算机的主要数据传输通道(例如总线),与其它部件交换信息。处

理器可以包括多个逻辑处理器(logical processor,LP),该逻辑处理器也可以称为物理处理器,例如LP1、LP2、...、LPn,n表示逻辑处理器的个数,n的取值可以为正整数,逻辑处理器是处理器socket中所包含的、具有独立执行线程的处理器单元。虚拟机可以包括多个虚拟处理器(virtual processor,VP),例如Vp1、VP2、...、VPn,n的取值可以为正整数。

[0063] 如图2,本申请实施例提供一种通信方法,包括:

[0064] 201、当第一逻辑处理器进入虚拟化模式之后,第一逻辑处理器获取IPI信息,IPI信息包括:目标虚拟处理器的标识和虚拟中断信息。

[0065] 本申请实施例中,第一逻辑处理器上当前运行有虚拟处理器,将第一逻辑处理器上正在运行的虚拟处理器定义为第一虚拟处理器,该第一虚拟处理器需要向目标虚拟处理器注入IPI时,第一虚拟处理器可以使用第一逻辑处理器来发送IPI信息,其中,该IPI信息包括:目标虚拟处理器的标识和虚拟中断信息。该目标虚拟处理器的标识可以是目标虚拟处理器的编号,该虚拟中断信息可以是虚拟IPI的详情信息,例如该虚拟中断信息可以包括中断号和中断发送方式。

[0066] 202、第一逻辑处理器对目标虚拟处理器的标识进行虚拟处理器地址翻译,得到目标虚拟处理器当前运行所在的目标逻辑处理器。

[0067] 在本申请实施例中,第一逻辑处理器中可以实现虚拟处理器编号到逻辑处理器编号的翻译,其功能包括虚拟处理器编号到逻辑处理器编号对应表的维护和查询。因此当第一逻辑处理器获取到目标虚拟处理器的标识之后,可以对目标虚拟处理器的标识进行虚拟处理器地址翻译,得到目标虚拟处理器当前运行所在的目标逻辑处理器。

[0068] 在本申请的一些实施例中,步骤202第一逻辑处理器对目标虚拟处理器的标识进行虚拟处理器地址翻译,包括:

[0069] 第一逻辑处理器通过查询虚拟处理器到逻辑处理器的路由表,对目标虚拟处理器的标识进行虚拟处理器地址翻译。

[0070] 其中,在每个逻辑处理器上运行虚拟处理器时,每个逻辑处理器都对应一个根据虚拟处理器编号进行索引的虚拟处理器到逻辑处理器的路由表,一个逻辑处理器可以访问自己对应的虚拟处理器到逻辑处理器的路由表,从而完成从虚拟处理器到相应的逻辑处理器的地址翻译。

[0071] 在本申请的一些实施例中,虚拟处理器到逻辑处理器的路由表存储在第一逻辑处理器的存储区域中;或,

[0072] 虚拟处理器到逻辑处理器的路由表存储在第一逻辑处理器所在的可用于物理插拔的处理器存储区域中;或,

[0073] 虚拟处理器到逻辑处理器的路由表存储到内存中。

[0074] 其中,可用于物理插拔的处理器是指前述的处理器socket。即虚拟处理器到逻辑处理器的路由表可以由一个逻辑处理器单独存储,也可以在该逻辑处理器所属的处理器socket的范围内进行全局共享,或者存储到内存中,从而每个逻辑处理器可以使用一个虚拟处理器指针从该内存中读取到虚拟处理器到逻辑处理器的路由表。

[0075] 在本申请的一些实施例中,第一逻辑处理器具有用于修改虚拟处理器到逻辑处理器的路由表的接口。

[0076] 其中,逻辑处理器还可以提供修改虚拟处理器到逻辑处理的路由表的支持机制,

例如在此支持机制被触发时,当前逻辑处理器对自身可访问的虚拟处理器到逻辑处理的路由表进行修改。又如,逻辑处理器可利用处理器间通信机制,通知其它逻辑处理器修改虚拟处理器到逻辑处理的路由表,具体实现方式此处不做限定。

[0077] 203、第一逻辑处理器向目标逻辑处理器发送IPI信息。

[0078] 在本申请实施例中,第一逻辑处理器确定出目标虚拟处理器当前运行所在的目标逻辑处理器之后,第一逻辑处理器可以向目标逻辑处理器发送IPI信息,使得该目标逻辑处理器可以按照IPI信息执行IPI注入。

[0079] 在本申请的一些实施例中,本申请实施例提供的通信方法还可以包括如下步骤:

[0080] 第一逻辑处理器对目标虚拟处理器的标识进行虚拟处理器地址翻译之后,确定目标虚拟处理器没有在任何一个逻辑处理器运行时,第一逻辑处理器存储IPI信息,并退出虚拟化模式;

[0081] 第一逻辑处理器通知VMM第一逻辑处理器存储了所述IPI信息。

[0082] 其中,第一逻辑处理器对目标虚拟处理器的标识进行虚拟处理器地址翻译之后,确定目标虚拟处理器没有在任何一个逻辑处理器运行时,则说明该第一逻辑处理器无法确定出目标虚拟处理器当前运行所在的目标逻辑处理器,此时可以将IPI信息存储下来,例如存储到发送方异常信息存储区域中,使得运行于非虚拟化模式的vmm软件可以查询到IPI发送详情信息。

[0083] 在本申请的一些实施例中,本申请实施例提供的通信方法还可以包括如下步骤:

[0084] 当第一逻辑处理器向目标逻辑处理器发送IPI信息失败时,第一逻辑处理器存储IPI信息,并退出虚拟化模式;

[0085] 第一逻辑处理器通知VMM第一逻辑处理器存储了IPI信息。

[0086] 其中,第一逻辑处理器可以设置“投送中断”时所使用的中断号,在逻辑处理器进入虚拟化模式后,执行IPI信息。退出虚拟化模式之后,则判断退出原因,若原因是“虚拟化模式下投送虚拟中断”,则由VMM使用中断号继续执行IPI中断信息。

[0087] 在本申请的一些实施例中,本申请实施例提供的通信方法还可以包括如下步骤:

[0088] 第一逻辑处理器向目标逻辑处理器发送目标虚拟处理器所在的虚拟机 (Virtual Machine, VM) 的标识。

[0089] 其中,第一逻辑处理器还需要向目标逻辑处理器发送目标虚拟处理器所在的VM的标识,以使得接收方可以确定出该目标虚拟处理器所在的VM。

[0090] 通过前述实施例的举例说明可知,当第一逻辑处理器进入虚拟化模式之后,第一逻辑处理器获取处理器中断IPI信息,IPI信息包括:目标虚拟处理器的标识和虚拟中断信息;第一逻辑处理器对目标虚拟处理器的标识进行虚拟处理器地址翻译,得到目标虚拟处理器当前运行所在的目标逻辑处理器;第一逻辑处理器向目标逻辑处理器发送IPI信息。本申请实施例中,第一逻辑处理器可以对目标虚拟处理器的标识进行虚拟处理器地址翻译,从而可以确定出目标虚拟处理器当前运行所在的目标逻辑处理器,因此第一逻辑处理器可以在虚拟化模式直接向正在运行的目标虚拟处理器发送IPI信息,使得目标虚拟处理器可以被注入IPI,不需要通过VMM来实现对IPI信息进行发送,第一逻辑处理器也不需要退出虚拟化模式,因此可以减少对发送方的处理器资源占用和IPI的最终传输延迟。

[0091] 前述实施例对发送方的逻辑处理器的执行流程进行了说明,接下来从接收方的逻

辑处理器的执行流程进行举例说明,请参阅图3所示,本申请实施例提供的通信方法主要包括如下步骤:

[0092] 301、当第二逻辑处理器进入虚拟化模式之后,第二逻辑处理器接收第一逻辑处理器发送的IPI信息,IPI信息包括:目标虚拟处理器的标识和虚拟中断信息。

[0093] 其中,本申请实施例中基于处理器硬件的通信机制,第一逻辑处理器发送IPI信息后,第二逻辑处理器作为接收方可以接收该IPI信息,第二逻辑处理器可以从该IPI信息确定出目标虚拟处理器的标识和虚拟中断信息。

[0094] 302、第二逻辑处理器根据虚拟中断信息对第二虚拟处理器进行虚拟中断注入处理。

[0095] 在本申请实施例中,正在运行的第二虚拟处理器就是需要被注入中断的虚拟处理器,由第二逻辑处理器根据虚拟中断信息对第二虚拟处理器进行虚拟中断注入处理。通过第一逻辑处理器和第二逻辑处理器之间的硬件直接通信,本申请实施例中可以实现对目标虚拟处理器的虚拟终端注入处理,整个过程不需要VMM软件的接入,第二逻辑处理器也不需要退出虚拟化模式。

[0096] 在本申请的一些实施例中,在步骤301执行之后,还可以执行如下步骤:

[0097] 第二逻辑处理器确定第二虚拟处理器的标识是否与目标虚拟处理器的标识相同,第二虚拟处理器是第二逻辑处理器进入虚拟模式后正在运行的虚拟处理器。

[0098] 当第二虚拟处理器的标识与目标虚拟处理器的标识相同时,触发执行前述的步骤302:第二逻辑处理器根据虚拟中断信息对第二虚拟处理器进行虚拟中断注入处理。

[0099] 在本申请实施例中,第二逻辑处理器判断该第二逻辑处理器当前正在运行的第二虚拟处理器是否就是该目标虚拟处理器,例如第二逻辑处理器可以确定第二虚拟处理器的标识是否与目标虚拟处理器的标识相同,第二虚拟处理器是第二逻辑处理器进入虚拟模式后正在运行的虚拟处理器。

[0100] 在本申请实施例中,当第二虚拟处理器的标识与目标虚拟处理器的标识相同时,说明该正在运行的第二虚拟处理器就是需要被注入中断的虚拟处理器,此时由第二逻辑处理器根据虚拟中断信息对第二虚拟处理器进行虚拟中断注入处理。通过第一逻辑处理器和第二逻辑处理器之间的硬件直接通信,本申请实施例中可以实现对目标虚拟处理器的虚拟终端注入处理,整个过程不需要VMM软件的接入,第二逻辑处理器也不需要退出虚拟化模式。

[0101] 在本申请的一些实施例中,本申请实施例提供的通信方法还可以包括如下步骤:

[0102] 当第二虚拟处理器的标识与目标虚拟处理器的标识不相同时,第二逻辑处理器存储IPI信息,并退出所述虚拟化模式。

[0103] 其中,第二逻辑处理器确定当前正在运行的第二虚拟处理器并不是目标虚拟处理器时,第二逻辑处理器可以将IPI信息存储下来,例如存储到接收方异常信息存储区域中,使得运行于非虚拟化模式的vmm软件可以查询到IPI发送详情信息。

[0104] 在本申请的一些实施例中,本申请实施例提供的通信方法还可以包括如下步骤:

[0105] 当第二逻辑处理器进入非虚拟化模式之后,第二逻辑处理器接收第一逻辑处理器发送的IPI信息,IPI信息包括:目标虚拟处理器的标识和虚拟中断信息;

[0106] 第二逻辑处理器确定第二虚拟处理器的标识是否与目标虚拟处理器的标识相同;

[0107] 当第二虚拟处理器的标识与目标虚拟处理器的标识相同时,第二逻辑处理器将第二逻辑处理器和第二虚拟处理器的对应关系更新到虚拟处理器到逻辑处理器的路由表中。

[0108] 其中,当第二逻辑处理器进入非虚拟化模式之后,即第二逻辑处理器并不是处于虚拟化模式,第二逻辑处理器可以将第二逻辑处理器和第二虚拟处理器的对应关系更新到虚拟处理器到逻辑处理器的路由表中,从而实现逻辑处理器对于该路由表的动态更新。

[0109] 通过前述实施例的举例说明可知,本申请实施例中,第一逻辑处理器可以对目标虚拟处理器的标识进行虚拟处理器地址翻译,从而可以确定出目标虚拟处理器当前运行所在的目标逻辑处理器,因此第一逻辑处理器可以在虚拟化模式直接向正在运行的目标虚拟处理器发送IPI信息,第二逻辑处理器作为目标逻辑处理器,该第二逻辑处理器上运行有该目标虚拟处理器,因此第二逻辑处理器可以对目标虚拟处理器进行IPI注入,不需要通过VMM来实现对IPI信息进行发送,第一逻辑处理器也不需要退出虚拟化模式,因此可以减少对发送方的处理器资源占用和IPI的最终传输延迟。

[0110] 为便于更好的理解和实施本申请实施例的上述方案,下面举例相应的应用场景来进行具体说明。

[0111] 本申请实施例可以解决的技术问题是HVM的虚拟处理器之间必须借助VMM发送IPI,而无法直接通信,从而导致CPU资源占用和IPI的最终传输延迟的问题。后续实施例中发送IPI和发送IPI中断的含义相同。

[0112] 虚拟处理器之间借助VMM发送IPI的过程中导致“CPU资源占用和IPI的最终传输延迟的问题”的主要原因是“借助VMM进行IPI通信”的过程中,逻辑处理器会退出虚拟化模式,具体原因如下:

[0113] 1)、发送IPI中断时需要指定接收方的逻辑处理器编号,但HVM感知到的虚拟处理器编号与物理计算机的逻辑处理器编号一般是不同的,若HVM采用标准的IPI发送逻辑,IPI将被送到错误的逻辑处理器;

[0114] 2)、若允许虚拟处理器直接发送IPI中断,则恶意的虚拟机可能会通过向虚拟机没有使用的逻辑处理器发送IPI,造成系统死机或影响其他虚拟机的正常工作;

[0115] 3)、逻辑处理器无法正确处理“接收虚拟IPI中断的虚拟处理器当前不在运行、甚至处于阻塞状态”的情形。

[0116] 本申请实施例中,逻辑处理器具有“可获得虚拟处理器到逻辑处理器对应关系”的机制。逻辑处理器具有在虚拟化模式下不借助VMM,通过目标逻辑处理器新增的“虚拟处理器地址翻译模块”监控目标虚拟处理器是否正在运行,从而判断是否需要VMM软件的介入。在逻辑处理器因虚拟处理器执行IPI发送指令而退出虚拟化模式时,具有向VMM软件暴露目标虚拟处理器编号和虚拟IPI中断详情的功能的机制,从而使得VMM软件可以在逻辑处理器退出虚拟化模式后,查询获得退出的原因,进而由VMM软件完成虚拟IPI的发送过程。

[0117] 需要说明的是,本申请实施例中,逻辑处理器发送IPI发送命令,不需要退出虚拟化模式。

[0118] 本申请实施例的应用场景是计算机系统的CPU,如x86处理器,ARM处理器等。

[0119] 图4为本申请实施例提供的逻辑处理器的基本架构示意图。本申请实施例主要对CPU硬件进行优化。如图4所示,本申请实施例针对CPU硬件,除了包含指令处理逻辑模块之外,还包括如下模块:虚拟处理器地址翻译模块,虚拟处理器中断收发模块,目标匹配模块,

发送方异常信息存储模块,接收方异常信息存储模块,发送方异常信息查询模块,接收方异常信息查询模块,接收方异常报告模块。

[0120] 接下来对上述各模块的功能及其相互关系进行描述如下:

[0121] 虚拟处理器地址翻译模块:负责实现虚拟处理器编号到逻辑处理器编号的翻译,其功能包括虚拟处理器编号到逻辑处理器编号对应表的维护和查询。当逻辑处理器运行虚拟化模式因执行IPI发送指令,通过虚拟处理器地址翻译模块无法获得有效的逻辑处理器编号时,逻辑处理器将待发送的IPI详情记录在发送方异常信息存储模块,并退出虚拟化模式;

[0122] 目标匹配模块:负责判断逻辑处理器接收到的虚拟中断注入请求的目标接收者是否是当前正在运行的虚拟处理器;

[0123] 虚拟处理器中断收发模块:包括发送端接口和中断注入子模块两部分。指令处理逻辑模块通过向当前逻辑处理器的发送端接口发送请求,来通知需接收IPI中断的逻辑处理器的中断注入子模块。当虚拟处理器中断收发模块的中断注入子模块接收到来自其它逻辑处理器的中断注入请求时,若当前逻辑处理器正处于非虚拟化模式,则触发接收方异常报告模块工作。若当前逻辑处理器正处于虚拟化模式,就根据目标匹配模块的判定结果,决定是向虚拟处理器投送中断。或者,当前逻辑处理器将接收到的中断注入请求详情记录在接收方异常信息存储模块,且转入非虚拟化模式。

[0124] 发送方异常信息查询模块和接收方异常信息查询模块,可以为运行于非虚拟化模式的VMM软件提供查询IPI发送详情的支持。

[0125] 接收方异常报告模块被触发后,执行VMM软件在此之前向逻辑处理器注册的异常处理流程。

[0126] 在本申请实施例所述的逻辑处理器中,位于逻辑处理器LP\_X的虚拟处理器VP\_x向虚拟处理器VP\_y发送IPI中断时,主要过程可以包括如下步骤:

[0127] 步骤一,主要包括如下1)和2)这两种情况。

[0128] 1)逻辑处理器LP\_X的虚拟处理器地址翻译模块显示“虚拟处理器VP\_y当前正在逻辑处理器LP\_Y上运行”。

[0129] 在虚拟处理器VP\_x执行“向虚拟处理器VP\_y发送IPI中断”的指令时,LP\_X的指令处理逻辑模块通过其虚拟处理器地址翻译模块,得到VP\_y位于有效的逻辑处理器LP\_Y上,由指令处理逻辑模块通知虚拟处理器中断收发模块的发送端接口进行中断的投送,或者由虚拟处理器地址翻译模块通知虚拟处理器中断收发模块的发送端接口进行中断的投送。逻辑处理器LP\_X的发送端接口将虚拟中断注入请求发送到逻辑处理器LP\_Y的中断注入子模块。

[0130] 2)逻辑处理器LP\_X的虚拟处理器地址翻译模块显示“虚拟处理器VP\_y当前没在逻辑处理器上运行”:

[0131] 指令处理逻辑模块通过虚拟处理器地址翻译模块获知“虚拟处理器VP\_y当前没有在逻辑处理器上运行”后,将IPI发送指令的详情写入发送方异常信息存储模块,该逻辑处理器LP\_X退出虚拟化模式。VMM软件可以通过发送方异常信息查询模块,获得指令处理逻辑模块在之前放入在发送方异常信息存储模块中的信息。

[0132] 步骤一执行之后,执行步骤二:

[0133] 当逻辑处理器LP\_Y的中断注入模块接收到来自其他逻辑处理器的中断注入消息时,主要包括如下1)和2)这两种情况。:

[0134] 1)、目标匹配模块判定“当前正在运行中断注入消息对应的虚拟处理器”。

[0135] 中断注入子模块在收到中断注入消息后,通过目标匹配模块发现当前正在运行与此中断注入消息对应的虚拟处理器,则直接通知指令处理逻辑模块对当前的虚拟处理器注入IPI中断。

[0136] 2)、目标匹配模块判定“当前未运行中断注入消息对应的虚拟处理器”。

[0137] 中断注入子模块在收到中断注入消息后,通过目标匹配模块发现应当由当前不在本逻辑处理器上运行的虚拟处理器接收此IPI中断,则将中断注入消息的详情记录到接收方异常信息存储模块,并通过接收方异常报告模块通知指令处理逻辑模块进行异常处理。指令处理逻辑模块在接收到此异常报告后,若当前正在虚拟化模式,则退出虚拟化模式;否则直接转入异常情况(如中断)处理流程。VMM软件可以通过接收方异常信息查询模块,获得指令处理逻辑模块在之前放入到接收方异常信息存储模块中的信息。

[0138] 需要指出的是,虚拟处理器地址翻译模块、虚拟处理器中断收发模块、目标匹配模块、接收方异常信息存储模块、发送方异常信息存储模块、接收方异常信息查询模块、发送方异常信息查询模块、接收方异常报告模块均是逻辑模块,两个或多个逻辑处理器完全可以共享其中的某个或某些逻辑模块的实现,比如:逻辑处理器LP\_X和逻辑处理器LP\_Y可能会共享虚拟处理器中断收发模块。

[0139] 本申请实施例提供的逻辑处理器具体可以是ARM处理器硬件。如图5所示,为本申请实施例的发送方的逻辑处理器和接收方的逻辑处理器之间的一种交互流程示意图。在IPI发送方的逻辑处理器进入虚拟化模式后,若获取到IPI发送指令时,则首先查找目标虚拟处理器所在的逻辑处理器,不能获得有效的逻辑处理器编号时,将IPI中断投送信息记录到VMM软件可访问的区域,然后退出虚拟化模式,并向VMM软件暴露退出原因。当能够获得有效的逻辑处理器时,利用中断信息传输机制,通知接收方逻辑处理器进行虚拟中断注入,其中,被传输的中断信息中还可以包含虚拟中断相关信息。接收方的逻辑处理器收到中断信息后,通过目标匹配模块的匹配逻辑判断当前是否正在运行目标虚拟处理器,若是,则进行虚拟中断注入,否则就转到异常处理。

[0140] 图6是本申请实施例中发送方的逻辑处理器和接收方的逻辑处理器之间的另一种交互流程示意图。主要包括如下过程:

[0141] 步骤一,设置“投送中断”时所使用的中断号,例如假设设置的中断号为中断vec。在逻辑处理器进入虚拟化模式后,执行IPI发送指令或其他指令,比如计算cpu指令、定时器指令。其中,该中断号用于区分不同虚拟处理器的中断。

[0142] 步骤二,若执行IPI发送指令后,退出虚拟化模式,则判断退出原因,若原因是“虚拟化模式下投送虚拟中断”,则使用中断号继续执行IPI中断发送指令;若是其他原因,则执行其他处理逻辑。

[0143] 步骤三,使用中断号继续执行IPI中断发送指令,判断之前在虚拟化模式下投送IPI是否失败,若是投送IPI失败,则根据当前的目标逻辑处理器更新虚拟处理器到逻辑处理器的路由信息,然后清除注入失败信息,最后进行虚拟中断的注入;若不是投送IPI失败,则直接进行虚拟中断的注入。



[0144] 图7为本申请实施例中处理器socket包括的模块示意图。处理器socket中所包含的主要相关模块：

[0145] 每个处理器socket都对应一个用来指示“本socket所包含的逻辑处理器编号”的位图lp\_bitmap。

[0146] 每个逻辑处理器都对应一个存储了<vmid, 虚拟处理器, ntr>的投送中断存储区域pintr\_info, 其中, 中断存储区域存储的“虚拟处理器”为虚拟IPI中断的目标虚拟处理器编号, vmid为目标虚拟处理器所在vm的编号, intr为被投送的虚拟IPI中断所包含的相关信息, 如虚拟IPI中断号、投送方式等。

[0147] 在运行虚拟处理器时, 每个逻辑处理器都对应一个存储了<虚拟处理器, vmid, intr>的投送失败存储区f\_info, 其中“虚拟处理器”为虚拟IPI中断的目标虚拟处理器编号, vmid为目标虚拟处理器所在的虚拟机的编号, intr为被投送的虚拟IPI中断所包含的相关信息, 如虚拟IPI中断号、投送方式。另外, 本申请实施例还具有向VMM软件暴露此信息的机制。

[0148] 在运行虚拟处理器时, 每个逻辑处理器都记录了当前虚拟处理器的编号和当前虚拟处理器所在VM的编号vmid。

[0149] 在运行虚拟处理器时, 每个逻辑处理器都对应一个根据虚拟处理器编号进行索引的虚拟处理器到逻辑处理器路由表, 逻辑处理器只能访问自己对应的虚拟处理器到逻辑处理器路由表。

[0150] 逻辑处理器提供“虚拟中断注入通知”的接收端硬件接口notify\_if, 接收端和发送端位于同一个处理器socket。

[0151] 逻辑处理器提供“虚拟中断注入通知转IPI中断号”的设置接口failed\_vec。

[0152] 逻辑处理器提供“接收虚拟中断注入消息”的硬件接口recv\_if, 接收端和发送端位于不同的处理器socket。虚拟IPI的发送方所在的lpcpu可通过此接口传递包括接收方所在的逻辑处理器编号、接收虚拟IPI的虚拟处理器编号、接收虚拟IPI的虚拟处理器所在的VM的编号、虚拟IPI的详情(如中断号、发送方式)；

[0153] 每个逻辑处理器都对应一个pintr\_info仲裁逻辑, 仲裁逻辑的输入为每个socket内其他逻辑处理器的pintr\_info请求线、本逻辑处理器的recv\_if处理逻辑, 如对于一个具有24个逻辑处理器的socket而言, 每个逻辑处理器的仲裁逻辑包含23条来自其它逻辑处理器的pintr\_info请求线、1条来自recv\_if处理逻辑的请求线。当pintr\_info仲裁逻辑同时接收到多个pintr\_info发送请求时, 按照轮流的方式, 放行某一个pintr\_info发送请求, 以免多个发送方同时写pintr\_info存储区造成信息丢失。

[0154] 需要说明的是, 投送失败存储区域f\_info、虚拟处理器到物理处理器(即逻辑处理器)路由表、虚拟处理器id、vmid与逻辑处理器并不一定是一一对应关系, 比如, 对于X86处理器来说, f\_info、虚拟处理器到逻辑处理器路由表、虚拟处理器id、vmid可以被定义在vmcs控制结构中, 从而在执行不同的虚拟处理器时, 由于切换了vmcs, 逻辑处理器自动换用不同的f\_info、虚拟处理器到逻辑处理器路由表、虚拟处理器id和vmid。另外, 处理器socket对应的lp\_bitmap可以由逻辑处理器在执行“设置逻辑处理器编号指令”时, 自动维护, 也可以由系统软件或虚拟化管理软件进行维护。

[0155] 如图8所示, 为本申请实施例提供的虚拟化模式下逻辑处理器执行IPI发送指令时

的流程示意图。在虚拟化模式运行的处理器按照如下方式执行IPI中断发送指令：

- [0156] 1)、若不能唯一确定IPI中断发送指令中指定的接收者,则转到步骤11)。
- [0157] 2)、根据虚拟处理器到物理处理器路由表,查找目标虚拟处理器所在的逻辑处理器lpcpu。
- [0158] 3)、若当前socket所对应的lp\_bitmap不包括步骤1)获得的lpcpu,则转到步骤8),若包括步骤1)得到的lpCPU,转到步骤4)。
- [0159] 4)、指令处理逻辑模块向lpcpu的pintr\_info仲裁逻辑发送请求,等待其放行信号。
- [0160] 5)、将当前逻辑处理器对应的vmid、IPI发送指令所指定的目标虚拟处理器编号、IPI发送指令所指定的IPI中断详情(如中断号、发送方式)放入lpcpu的投送中断存储区pintr\_info。步骤5执行之后,执行步骤9)。
- [0161] 6)、利用notify\_if,通知lpcpu进行虚拟IPI中断的注入。
- [0162] 7)、IPI中断发送指令执行结束。
- [0163] 8)、若lpcpu不是合法的逻辑处理器编号,则转到步骤11)。
- [0164] 9)、利用recv\_if,向目标lpcpu发送虚拟中断注入消息。
- [0165] 10)、IPI中断发送指令执行结束。
- [0166] 11)、将目标虚拟处理器编号(或编号的集合)、IPI发送指令所指定的IPI中断详情记入当前的投送失败存储区f\_info,且将f\_info中的vmid设置为无效值。
- [0167] 12)、退出虚拟化模式。

[0168] 如图9所示,为本申请实施例提供的虚拟化模式下逻辑处理器检测到notify\_if中有信息后的处理流程示意图。在虚拟化模式运行的逻辑处理器通过notify\_if接收到虚拟中断注入通知时,执行包括如下步骤的流程:

- [0169] 1)、将本逻辑处理器对应的pintr\_info信息缓存到cached\_info临时存储区中;
- [0170] 2)、若当前逻辑处理器对应的虚拟处理器id不等于cached\_info中记录的虚拟处理器编号,或当前逻辑处理器对应的vmid不等于cached\_info中记录的vmid,则转到步骤5);
- [0171] 3)、按照cached\_info记录的信息,向当前虚拟处理器注入虚拟中断;
- [0172] 4)、通知pintr\_info仲裁逻辑放行下一个pintr\_info发送请求,并结束处理过程;
- [0173] 5)、根据cached\_info中的信息,填写本逻辑处理器的f\_info,其vmid为cached\_info中的对应值;
- [0174] 6)、通知pintr\_info仲裁逻辑放行下一个pintr\_info发送请求,并退出虚拟化运行模式,结束对于“虚拟中断注入通知”的处理。

[0175] 需要说明的是,在运行虚拟处理器前,VMM利用failed\_vec接口,设置notify\_vector:“虚拟中断注入失败后,通过notify\_if接收到虚拟中断注入通知,或者通过recv\_if接收到虚拟中断注入消息的逻辑处理器所接收到的IPI中断的编号”。

[0176] 接下来针对notify\_if和recv\_if这两种实现场景对接收端的逻辑处理器的执行流程进行举例说明,在非虚拟化模式运行的逻辑处理器通过notify\_if接收到虚拟中断注入通知时,执行包括如下步骤的流程:

- [0177] 1)、将本逻辑处理器对应的pintr\_info信息缓存到cached\_info临时存储区中。

[0178] 2)、根据cached\_info中的信息,填写本逻辑处理器的f\_info,其vmid为cached\_info中的对应值。

[0179] 3)、通知pintr\_info仲裁逻辑可以放行下一个pintr\_info发送请求。

[0180] 4)、执行notify\_vector所对应的中断处理程序。

[0181] 5)、结束对于虚拟中断注入通知的处理。

[0182] 逻辑处理器通过recv\_if接收到虚拟中断注入消息时,执行包括如下步骤的流程:

[0183] 1)、将消息中包含的目的虚拟处理器编号、目的vm编号,IPI中断详情放入cached\_info临时存储区域。

[0184] 2)、向虚拟中断注入消息中指定的目的逻辑处理器的pintr\_info仲裁逻辑发送请求,等待其放行信号。

[0185] 3)、将cached\_info中所包含的信息放入lpcpu的投送中断存储区pintr\_info。

[0186] 4)、利用notify\_if,通知当前逻辑处理器进行虚拟IPI中断的注入。

[0187] 5)、结束对于虚拟中断注入消息的处理。

[0188] 运行于在非虚拟化模式下的逻辑处理器在执行IPI中断发送指令时,发现此指令的执行将导致某个虚拟处理器接收到虚拟IPI中断,则在按照标准方式执行此IPI发送指令之外,还额外执行包括如下步骤的流程:

[0189] 1)、若当前逻辑处理器的f\_info信息中包含的vmid为有效值,则处理结束。否则继续下面的过程。

[0190] 2)、获得当前IPI中断发送指令所对应的逻辑处理器编号dest\_cpu。

[0191] 3)、根据当前逻辑处理器的f\_info信息,获得目标虚拟处理器编号为dest\_虚拟处理器。

[0192] 4)、更新当前逻辑处理器的虚拟处理器到逻辑处理器路由表中,dest\_虚拟处理器所对应的逻辑处理器为dest\_cpu。

[0193] 5)、处理结束。

[0194] 本申请的前述实施例中,逻辑处理器都包含一个虚拟处理器到逻辑处理器路由表,该虚拟处理器到逻辑处理器路由表对VMM软件透明,即VMM软件是无感知的。在虚拟化模式下,逻辑处理器执行IPI发送指令时,会根据此路由表,将目标虚拟处理器编号翻译为逻辑处理器编号,并向翻译后得到的逻辑处理器提交虚拟IPI中断注入通知。逻辑处理器在收到虚拟IPI中断注入通知后,在确认当前正在运行此虚拟IPI中断对应的目标虚拟处理器时,才向当前虚拟处理器注入虚拟IPI中断;在非虚拟化模式下逻辑处理器在执行IPI发送指令时,会自动填充虚拟处理器到逻辑处理器路由表中的相应表项。

[0195] 本申请实施例可以保证逻辑处理器可以在虚拟化模式直接向某个正在运行的虚拟处理器发送IPI中断,且能够处理“目标虚拟处理器当前实际并未运行”的情形。

[0196] 在本申请实施例中,发送方的逻辑处理器可以根据路由逻辑仅能获得目标虚拟处理器所在的逻辑处理器,并不直接操纵目标虚拟处理器的中断描述信息。这种方式的优点是:1)在路由表非全局共享时,本申请实施例可以实现自适应的构建路由表。2)在虚拟中断投送时,逻辑处理器之间只有一种信息传输通道(即中断投送信息),因此无需针对多种类型信息传递时需要解决的信息同步问题,简化了两个逻辑处理器之间的处理过程。

[0197] 本申请的另一一些实施例,VM的虚拟处理器到逻辑处理器路由表不再是逻辑处理器

私有的数据结构,而是一个由整个处理器socket内所有的逻辑处理器共享使用全局数据结构。举例说明,根据<vm编号,虚拟处理器编号>的散列值,将<vm编号,虚拟处理器编号,逻辑处理器编号>存储在socket内的存储区域(如cache中)。当逻辑处理器加载运行虚拟处理器时(比如x86处理器加载虚拟处理器对应的VMCS数据结构时),逻辑处理器根据当前虚拟处理器的编号、当前虚拟处理器所在虚拟机的编号、当前逻辑处理器编号,更新散列表。当逻辑处理器卸载虚拟处理器时(比如x86处理器卸载虚拟处理器对应的VMCS数据结构时),逻辑处理器根据当前虚拟处理器的编号、当前虚拟处理器所在虚拟机的编号、当前逻辑处理器编号,将散列表中<vmid,虚拟处理器id>对应的表项删除。

[0198] 本申请的前述实施例中,逻辑处理器可以避免存放虚拟处理器到逻辑处理器路由表,且路由表的构建速度更快。

[0199] 本申请的另一一些实施例,虚拟处理器到逻辑处理器路由表被存放在内存中,在运行虚拟处理器时,每个逻辑处理器都对应一个虚拟处理器指针,该指针指向了虚拟处理器到逻辑处理器路由表在内存中的位置。虚拟处理器到逻辑处理器路由表与逻辑处理器并不是一一对应关系,比如,对于X86处理器来说,虚拟处理器指针可以被定义在vmcs结构中,从而在执行不同的虚拟处理器时,由于切换了vmcs,逻辑处理器自动换用不同的虚拟处理器子帧。

[0200] 当虚拟处理器到逻辑处理器路由表被存放在允许处理器进行缓存的内存区域时,处理器可以利用其高速缓存机制,实现对于虚拟处理器到逻辑处理器路由表的快速检索。

[0201] 当逻辑处理器加载运行虚拟处理器时(比如x86处理器加载虚拟处理器对应的VMCS数据结构时),逻辑处理器直接设置虚拟处理器指针(即虚拟处理器\_rte)所指向的虚拟处理器到逻辑处理器路由表。比如:若虚拟处理器子帧的值为0x10000、路由表中每个表项占用2个字节,当逻辑处理器10在加载编号为80的虚拟处理器时,逻辑处理器10将把内存0x100a0处的2个字节设置为10。

[0202] 当逻辑处理器卸载虚拟处理器时(比如x86处理器卸载虚拟处理器对应的VMCS数据结构时),逻辑处理器直接清除虚拟处理器到逻辑处理器路由表中的相应表项。比如:若虚拟处理器指针的值为0x10000、路由表中每个表项占用2个字节,当逻辑处理器10在卸载编号为80的虚拟处理器时,逻辑处理器10将把内存0x100a0处的2个字节设置为无效值(如0xffff)。

[0203] 本申请的另一一些实施例中,pintr\_info中包含了src\_cpu信息,其中src\_cpu指示了“执行虚拟IPI发送操作的逻辑处理器编号”,比如,若虚拟处理器20当前运行编号为10的逻辑处理器上、虚拟处理器20在虚拟化模式下向虚拟处理器30发送IPI中断、当前虚拟处理器到逻辑处理器路由表指示“虚拟处理器30位于编号为11的逻辑处理器上”、且当前无其它逻辑处理器向编号为11的逻辑处理器投送中断,则虚拟处理器20本次IPI发送动作将导致编号为11的逻辑处理器的pintr\_info的src\_cpu变为10。

[0204] 如图10所示,为本申请实施例提供的虚拟化模式下逻辑处理器对于notify\_if请求的处理流程示意图。在虚拟化模式运行的逻辑处理器通过notify\_if接收到虚拟中断注入通知时,执行包括如下步骤的流程:

[0205] 1)、将本逻辑处理器对应的pintr\_info信息缓存到cached\_info临时存储区中;

[0206] 2)、若cached\_info中包含的虚拟处理器编号小于0,则转到步骤15)。其中,虚拟处

理器编号小于0,表示该虚拟处理器不合法。

[0207] 3)、若当前逻辑处理器对应的虚拟处理器id不等于cached\_info中记录的虚拟处理器编号,或当前逻辑处理器对应的vmid不等于cached\_info中记录的vmid,则转到步骤6)。

[0208] 4)、按照cached\_info记录的信息,向当前虚拟处理器注入虚拟中断。

[0209] 5)、通知pintr\_info仲裁逻辑放行下一个pintr\_info发送请求,并完成对于“虚拟中断注入通知”的处理。

[0210] 6)、根据cached\_info中的信息,填写本逻辑处理器的f\_info,其vmid为cached\_info中的对应值。

[0211] 7)、获得cached\_info中包含的src\_cpu。

[0212] 8)、若src\_cpu不位于本socket,转到步骤9),否则转到步骤11)。

[0213] 9)、向src\_cpu发送recv\_if消息,消息体包含的虚拟处理器编号cached\_info中虚拟处理器编号的按“位”取反后的值。如:若处理器最大只能支持32767个虚拟处理器时,当cached\_info中虚拟处理器编号为1,此recv\_if中虚拟处理器编号为0xffffe,vm编号为cached\_info中的编号。

[0214] 10)、通知pintr\_info仲裁逻辑放行下一个pintr\_info发送请求,并结束处理过程。

[0215] 11)、向src\_cpu的pintr\_info仲裁逻辑发送请求,等待其放行信号。

[0216] 12)、将cached\_info中包含的vmid、cached\_info中虚拟处理器编号按“位”取反后的值放入src\_cpu的投送中断存储区pintr\_info。

[0217] 13)、利用notify\_if,通知src\_cpu进行虚拟IPI中断的注入。

[0218] 14)、通知pintr\_info仲裁逻辑放行下一个pintr\_info发送请求、退出虚拟化运行模式,并结束对于“虚拟中断注入通知”的处理。

[0219] 15)、将cached\_info中所包含的虚拟处理器编号按“位”取反,获得真实的虚拟处理器编号r虚拟处理器。

[0220] 16)、若当前逻辑处理器对应的vmid与cached\_info中指定的虚拟机编号相等,则将逻辑处理器所使用的虚拟处理器到逻辑处理器路由表中,r虚拟处理器对应的表项设置为无效值。

[0221] 17)、通知pintr\_info仲裁逻辑放行下一个pintr\_info发送请求,并完成对于虚拟中断注入通知的处理。

[0222] 在非虚拟化模式运行的逻辑处理器通过notify\_if接收到虚拟中断注入通知时,执行包括如下步骤的流程:

[0223] 1)、将本逻辑处理器对应的pintr\_info信息缓存到cached\_info临时存储区中。

[0224] 2)、若cached\_info中包含的虚拟处理器编号小于0,则转到步骤5)。

[0225] 3)、根据cached\_info中的信息,填写本逻辑处理器的f\_info,其vmid为cached\_info中的对应值。

[0226] 4)、通知pintr\_info仲裁逻辑可以放行下一个pintr\_info发送请求,并执行notify\_vector所对应的中断处理程序,且结束对于虚拟中断注入通知的处理;

[0227] 5)、将cached\_info中所包含的虚拟处理器编号按“位”取反,获得真实的虚拟处理

器编号r虚拟处理器。

[0228] 6)、若逻辑处理器对应的vmid与cached\_info中指定的虚拟机编号相同,则将逻辑处理器所使用的虚拟处理器到逻辑处理器路由表中,r虚拟处理器对应的表项设置为无效值。

[0229] 7)、通知pintr\_info仲裁逻辑放行下一个pintr\_info发送请求,完成对于虚拟中断注入通知的处理。

[0230] 本申请的另一些实施例中,逻辑处理器还提供修改虚拟处理器到物理处理器路由表的支持机制(如指令、特定寄存器等)。例如一种可行的实现方式是,在此支持机制被触发时,当前逻辑处理器对自身可访问的虚拟处理器到物理处理器路由表进行修改。又如另一种可行的实现方式是:逻辑处理器可利用处理器间通信机制,通知其它逻辑处理器修改虚拟处理器到逻辑处理器路由表。

[0231] 本申请实施例可以给软件更大的自由度,方便进行虚拟处理器到逻辑处理器路由表的构建。相对于前述实施例,本申请实施例可以由软件实现非频繁出现的“需要通知IPI发送方更新虚拟处理器到逻辑处理器路由表”情形,从而简化处理器的硬件设计。

[0232] 本申请实施例可以包括如下的技术方案:

[0233] (一)逻辑处理器具有“可获得虚拟处理器到逻辑处理器对应关系”的机制,即逻辑处理器新增配置“根据虚拟处理器编号查询此虚拟处理器当前所在的逻辑处理器编号”的功能,即新增“虚拟处理器地址翻译模块”等功能,发送方根据路由逻辑仅能获得目标虚拟处理器所在的逻辑处理器,并不直接操纵目标虚拟处理器的中断描述信息:

[0234] 1、可以是逻辑处理器内部私有的虚拟处理器到逻辑处理器路由表;

[0235] 2、可以是被socket内所有逻辑处理器共享的<vm标识,虚拟处理器标识>到逻辑处理器路由表;

[0236] 3、可以是存放在内存中的、对应每个虚拟机一份的、虚拟处理器到逻辑处理器路由表,且逻辑处理器内部记录了此虚拟机唯一路由表的内存地址;

[0237] 4、多个逻辑处理器之间的“虚拟处理器地址翻译模块”等可以共享;

[0238] 5、在路由表非全局共享时,本方式可以实现自适应的构建路由表(见实施例1的说明);

[0239] 6、在虚拟中断投送时,逻辑处理器之间只有一种信息传输通道(中断投送信息)。

[0240] (二)逻辑处理器具有在虚拟化模式下不借助VMM,通过获得的目标逻辑处理器新增的“虚拟处理器地址翻译模块”监控目标虚拟处理器是否正在运行,从而判断是否需要VMM软件的介入:

[0241] 1、若判断显示当前目标虚拟处理器正在运行,发送方通过新增的“虚拟处理器中断收发模块”的“发送端接口”子模块将虚拟IPI中断注入请求发送到目标逻辑处理器中新增的“虚拟处理器中断收发模块”的“中断注入”子模块:

[0242] 当目标逻辑处理器的中断注入子模块接收到来自其他逻辑处理器的中断注入消息时,新增的“目标匹配模块”判定当前是否正在运行“中断注入消息对应的虚拟处理器”:

[0243] 1)若判断正在运行,则直接通知指令处理逻辑模块对当前的虚拟处理器注入IPI中断,完成IPI中断直通;

[0244] 2)若判断未在运行,则将中断注入消息的详情记录到接收方新增的“异常信息存

储模块”，并通过接收方新增的“异常报告模块”通知指令处理逻辑模块进行异常处理。指令处理逻辑模块在接收到此异常报告后，若当前正在虚拟化模式，则退出虚拟化模式；否则直接转入异常情况(如中断)处理流程。

[0245] 2、若判断显示当前目标虚拟处理器处于未运行或阻塞状态，将IPI发送指令的详情写入发送方异常信息存储模块，并退出虚拟化模式。VMM软件可以通过发送方异常信息查询模块，获得指令处理逻辑模块在之前放入在发送方异常信息存储模块中的信息。

[0246] 3、多个逻辑处理器之间的“虚拟处理器中断收发模块”等可以共享。

[0247] (三)在逻辑处理器因虚拟处理器执行IPI发送指令而退出虚拟化模式时，参考如上(一)(二)机制，逻辑处理器通过新增配置“发送方异常信息存储/查询模块”，“接收方异常报告模块”、“接收方异常信息存储/查询模块”等功能，具有向VMM软件暴露目标虚拟处理器编号和虚拟IPI中断详情的功能，从而使得VMM软件可以在逻辑处理器退出虚拟化模式后，查询获得退出的原因，进而采用传统模式完成虚拟IPI的发送过程。

[0248] 首先，本申请实施例解决如下的问题：发送IPI中断时需要指定接收方的逻辑处理器编号，但HVM无法获得目标处理器编号，若HVM采用标准的IPI发送逻辑，IPI将被送到错误的逻辑处理器。

[0249] 本申请实施例中，逻辑处理器通过新增的“虚拟处理器地址翻译模块”具有“可获得虚拟处理器到逻辑处理器对应关系”的机制；

[0250] 本申请实施例中，HVM可通过“虚拟处理器地址翻译模块”获得目标逻辑处理器的编号。发送方根据路由逻辑仅能获得目标虚拟处理器所在的逻辑处理器，并不直接操纵目标虚拟处理器的中断描述信息。这种方式的优点是：1)在路由表非全局共享时，本方式可以实现自适应的构建路由表。2)在虚拟中断投送时，逻辑处理器之间只有一种信息传输通道(中断投送信息)。

[0251] 其次，现有技术的虚拟处理器必须借助VMM发送IPI中断，若允许直接发送IPI中断，则恶意的虚拟机可能会通过向虚拟机没有使用的逻辑处理器发送IPI中断，造成系统死机或影响其他虚拟机的正常工作；

[0252] 本申请实施例中，逻辑处理器具有在虚拟化模式下不借助VMM，通过获得的目标逻辑处理器新增的“虚拟处理器地址翻译模块”监控目标虚拟处理器是否正在运行，从而判断是否需要VMM软件的介入；

[0253] 本申请实施例中，若通过新增的“虚拟处理器地址翻译模块”监控判断目标虚拟处理器正在运行，则发出虚拟IPI中断注入请求，并且目标逻辑处理器新增的“目标匹配模块”判定当前是否正在运行“中断注入消息对应的虚拟处理器”，则可以完成HVM的虚拟处理器之间直接IPI通信。

[0254] 最后，现有技术的虚拟处理器借助VMM发送IPI中断的方法，导致逻辑处理器无法正确处理“接收虚拟IPI中断的虚拟处理器当前不在运行、甚至处于阻塞状态”的情形；

[0255] 本申请实施例中，在逻辑处理器因虚拟处理器执行IPI发送指令而退出虚拟化模式时，具有向VMM软件暴露目标虚拟处理器编号和虚拟IPI中断详情的功能的机制；

[0256] 本申请实施例中，使得VMM软件可以在逻辑处理器退出虚拟化模式后，查询获得退出的原因，进而采用传统模式完成虚拟IPI的发送过程。

[0257] 需要说明的是，对于前述的各方法实施例，为了简单描述，故将其都表述为一系列

的动作组合,但是本领域技术人员应该知悉,本申请并不受所描述的动作顺序的限制,因为依据本申请,某些步骤可以采用其他顺序或者同时进行。其次,本领域技术人员也应该知悉,说明书中所描述的实施例均属于优选实施例,所涉及的动作和模块并不一定是本申请所必须的。

[0258] 为便于更好的实施本申请实施例的上述方案,下面还提供用于实施上述方案的相关装置。

[0259] 本申请实施例还提供的一种逻辑处理器,该逻辑处理器具体为前述的第一逻辑处理器,请参阅图11所示,本申请实施例提供的第一逻辑处理器1100,可以包括:处理模块1101、发送模块1102,其中,

[0260] 处理模块1101,用于当第一逻辑处理器进入虚拟化模式之后,获取处理器中断IPI信息,所述IPI信息包括:目标虚拟处理器的标识和虚拟中断信息;

[0261] 所述处理模块1101,还用于对所述目标虚拟处理器的标识进行虚拟处理器地址翻译,得到所述目标虚拟处理器当前运行所在的目标逻辑处理器;

[0262] 发送模块1102,用于向所述目标逻辑处理器发送所述IPI信息。

[0263] 本申请实施例中,第一逻辑处理器上当前运行有虚拟处理器,将第一逻辑处理器上正在运行的虚拟处理器定义为第一虚拟处理器,该第一虚拟处理器需要向目标虚拟处理器注入IPI时,第一虚拟处理器可以使用第一逻辑处理器来发送IPI信息,其中,该IPI信息包括:目标虚拟处理器的标识和虚拟中断信息。该目标虚拟处理器的标识可以是目标虚拟处理器的编号,该虚拟中断信息可以是虚拟IPI的详情信息,例如该虚拟中断信息可以包括中断号和中断发送方式。

[0264] 在本申请实施例中,第一逻辑处理器中可以实现虚拟处理器编号到逻辑处理器编号的翻译,其功能包括虚拟处理器编号到逻辑处理器编号对应表的维护和查询。因此当第一逻辑处理器获取到目标虚拟处理器的标识之后,可以对目标虚拟处理器的标识进行虚拟处理器地址翻译,得到目标虚拟处理器当前运行所在的目标逻辑处理器。

[0265] 在本申请实施例中,第一逻辑处理器确定出目标虚拟处理器当前运行所在的目标逻辑处理器之后,第一逻辑处理器可以向目标逻辑处理器发送IPI信息,使得该目标逻辑处理器可以按照IPI信息执行IPI注入。

[0266] 在本申请的一些实施例中,所述处理模块1101,还用于对所述目标虚拟处理器的标识进行虚拟处理器地址翻译之后,确定所述目标虚拟处理器没有在任何一个逻辑处理器运行时,存储所述IPI信息,并退出所述虚拟化模式;

[0267] 所述发送模块102,还用于通知虚拟机监控器VMM所述第一逻辑处理器存储了所述IPI信息。

[0268] 其中,第一逻辑处理器对目标虚拟处理器的标识进行虚拟处理器地址翻译之后,确定目标虚拟处理器没有在任何一个逻辑处理器运行时,则说明该第一逻辑处理器无法确定出目标虚拟处理器当前运行所在的目标逻辑处理器,此时可以将IPI信息存储下来,例如存储到发送方异常信息存储区域中,使得运行于非虚拟化模式的vmm软件可以查询到IPI发送详情信息。

[0269] 在本申请的一些实施例中,所述处理模块1101,还用于当所述第一逻辑处理器向所述目标逻辑处理器发送所述IPI信息失败时,存储所述IPI信息,并退出所述虚拟化模式;



[0270] 所述发送模块102,还用于通知VMM所述第一逻辑处理器存储了所述IPI信息。

[0271] 其中,第一逻辑处理器可以设置“投送中断”时所使用的中断号,在逻辑处理器进入虚拟化模式后,执行IPI信息。退出虚拟化模式之后,则判断退出原因,若原因是“虚拟化模式下投送虚拟中断”,则由VMM使用中断号继续执行IPI中断信息。

[0272] 在本申请的一些实施例中,所述发送模块1102,还用于向所述目标逻辑处理器发送所述目标虚拟处理器所在的虚拟机VM的标识。

[0273] 其中,第一逻辑处理器还需要向目标逻辑处理器发送目标虚拟处理器所在的VM的标识,以使得接收方可以确定出该目标虚拟处理器所在的VM。

[0274] 在本申请的一些实施例中,所述处理模块1101,具体用于通过查询虚拟处理器到逻辑处理器的路由表,对所述目标虚拟处理器的标识进行虚拟处理器地址翻译。

[0275] 其中,在每个逻辑处理器上运行虚拟处理器时,每个逻辑处理器都对应一个根据虚拟处理器编号进行索引的虚拟处理器到逻辑处理器的路由表,一个逻辑处理器可以访问自己对应的虚拟处理器到逻辑处理器的路由表,从而完成从虚拟处理器到相应的逻辑处理器的地址翻译。

[0276] 在本申请的一些实施例中,所述虚拟处理器到逻辑处理器的路由表存储在所述第一逻辑处理器的存储区域中;或,

[0277] 所述虚拟处理器到逻辑处理器的路由表存储在所述第一逻辑处理器所在的可用于物理插拔的处理器存储区域中;或,

[0278] 所述虚拟处理器到逻辑处理器的路由表存储到内存中。

[0279] 其中,可用于物理插拔的处理器是指前述的处理器socket。即虚拟处理器到逻辑处理器的路由表可以由一个逻辑处理器单独存储,也可以在该逻辑处理器所属的处理器socket的范围内进行全局共享,或者存储到内存中,从而每个逻辑处理器可以使用一个虚拟处理器指针从该内存中读取到虚拟处理器到逻辑处理器的路由表。

[0280] 在本申请的一些实施例中,所述第一逻辑处理器具有用于修改所述虚拟处理器到逻辑处理器的路由表的接口。

[0281] 其中,逻辑处理器还可以提供修改虚拟处理器到逻辑处理的路由表的支持机制,例如在此支持机制被触发时,当前逻辑处理器对自身可访问的虚拟处理器到逻辑处理的路由表进行修改。又如,逻辑处理器可利用处理器间通信机制,通知其它逻辑处理器修改虚拟处理器到逻辑处理的路由表,具体实现方式此处不做限定。

[0282] 通过前述实施例的举例说明可知,当第一逻辑处理器进入虚拟化模式之后,第一逻辑处理器获取处理器中断IPI信息,IPI信息包括:目标虚拟处理器的标识和虚拟中断信息;第一逻辑处理器对目标虚拟处理器的标识进行虚拟处理器地址翻译,得到目标虚拟处理器当前运行所在的目标逻辑处理器;第一逻辑处理器向目标逻辑处理器发送IPI信息。本申请实施例中,第一逻辑处理器可以对目标虚拟处理器的标识进行虚拟处理器地址翻译,从而可以确定出目标虚拟处理器当前运行所在的目标逻辑处理器,因此第一逻辑处理器可以在虚拟化模式直接向正在运行的目标虚拟处理器发送IPI信息,使得目标虚拟处理器可以被注入IPI,不需要通过VMM来实现对IPI信息进行发送,第一逻辑处理器也不需要退出虚拟化模式,因此可以减少对发送方的处理器资源占用和IPI的最终传输延迟。

[0283] 本申请实施例还提供的另一种逻辑处理器,该逻辑处理器具体为前述的第二逻辑

处理器,请参阅图12所示,本申请实施例提供的第二逻辑处理器1200,可以包括:接收模块1201、处理模块1202,其中,

[0284] 接收模块1201,用于当第二逻辑处理器进入虚拟化模式之后,接收第一逻辑处理器发送的IPI信息,所述IPI信息包括:目标虚拟处理器的标识和虚拟中断信息;

[0285] 处理模块1202,用于根据所述虚拟中断信息对所述第二虚拟处理器进行虚拟中断注入处理,所述第二虚拟处理器是所述第二逻辑处理器进入虚拟模式后正在运行的虚拟处理器。

[0286] 其中,本申请实施例中基于处理器硬件的通信机制,第一逻辑处理器发送IPI信息后,第二逻辑处理器作为接收方可以接收该IPI信息,第二逻辑处理器可以从该IPI信息确定出目标虚拟处理器的标识和虚拟中断信息。

[0287] 在本申请实施例中,正在运行的第二虚拟处理器就是需要被注入中断的虚拟处理器,由第二逻辑处理器根据虚拟中断信息对第二虚拟处理器进行虚拟中断注入处理。通过第一逻辑处理器和第二逻辑处理器之间的硬件直接通信,本申请实施例中可以实现对目标虚拟处理器的虚拟终端注入处理,整个过程不需要VMM软件的接入,第二逻辑处理器也不需要退出虚拟化模式。

[0288] 在本申请的一些实施例中,所述处理模块1202,还用于当所述第二虚拟处理器的标识与所述目标虚拟处理器的标识不相同,存储所述IPI信息,并退出所述虚拟化模式。

[0289] 在本申请的一些实施例中,所述处理模块1202,还用于所述接收模块1201接收第一逻辑处理器发送的IPI信息之后,确定第二虚拟处理器的标识是否与所述目标虚拟处理器的标识相同;

[0290] 所述处理模块1202,还用于当所述第二虚拟处理器的标识与所述目标虚拟处理器的标识相同时,触发执行如下步骤:根据所述虚拟中断信息对第二虚拟处理器进行虚拟中断注入处理。

[0291] 在本申请实施例中,第二逻辑处理器判断该第二逻辑处理器当前正在运行的第二虚拟处理器是否就是该目标虚拟处理器,例如第二逻辑处理器可以确定第二虚拟处理器的标识是否与目标虚拟处理器的标识相同,第二虚拟处理器是第二逻辑处理器进入虚拟模式后正在运行的虚拟处理器。

[0292] 通过前述实施例的举例说明可知,本申请实施例中,第一逻辑处理器可以对目标虚拟处理器的标识进行虚拟处理器地址翻译,从而可以确定出目标虚拟处理器当前运行所在的目标逻辑处理器,因此第一逻辑处理器可以在虚拟化模式直接向正在运行的目标虚拟处理器发送IPI信息,第二逻辑处理器作为目标逻辑处理器,该第二逻辑处理器上运行有该目标虚拟处理器,因此第二逻辑处理器可以对该目标虚拟处理器进行IPI注入,不需要通过VMM来实现对IPI信息进行发送,第一逻辑处理器也不需要退出虚拟化模式,因此可以减少对发送方的处理器资源占用和IPI的最终传输延迟。

[0293] 需要说明的是,上述装置各模块/单元之间的信息交互、执行过程等内容,由于与本申请方法实施例基于同一构思,其带来的技术效果与本申请方法实施例相同,具体内容可参见本申请前述所示的方法实施例中的叙述,此处不再赘述。

[0294] 本申请实施例还提供一种计算机存储介质,其中,该计算机存储介质存储有程序,该程序执行包括上述方法实施例中记载的部分或全部步骤。

[0295] 需要说明的是,本申请实施例中处理器还可以称为中央处理单元(英文全称:Central Processing Unit,英文简称:CPU)。具体的应用中,处理器的各个组件通过总线系统耦合在一起。处理器可以是一种集成电路芯片,具有信号的处理能力。在实现过程中,上述方法的各步骤可以通过处理器中的硬件的集成逻辑电路或者软件形式的指令完成。上述的处理器可以是通用处理器、数字信号处理器(英文全称:digital signal processing,英文缩写:DSP)、专用集成电路(英文全称:Application Specific Integrated Circuit,英文缩写:ASIC)、现场可编程门阵列(英文全称:Field-Programmable Gate Array,英文缩写:FPGA)或者其他可编程逻辑器件、分立门或者晶体管逻辑器件、分立硬件组件。可以实现或者执行本申请实施例中的公开的各方法、步骤及逻辑框图。通用处理器可以是微处理器或者该处理器也可以是任何常规的处理器等。结合本申请实施例所公开的方法的步骤可以直接体现为硬件译码处理器执行完成,或者用译码处理器中的硬件及软件模块组合执行完成。

[0296] 另外需说明的是,以上所描述的装置实施例仅仅是示意性的,其中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或者也可以不是物理单元,即可以位于一个地方,或者也可以分布到多个网络单元上。可以根据实际的需要选择其中的部分或者全部模块来实现本申请实施例方案的目的。另外,本申请提供的装置实施例附图中,模块之间的连接关系表示它们之间具有通信连接,具体可以实现为一条或多条通信总线或信号线。

[0297] 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本申请可以用硬件实现,或固件实现,或它们的组合方式来实现。当使用软件实现时,可以将上述功能存储在计算机可读介质中或作为计算机可读介质上的一个或多个指令或代码进行传输。计算机可读介质包括计算机存储介质和通信介质,其中通信介质包括便于从一个地方向另一个地方传送计算机程序的任何介质。存储介质可以是计算机能够存取的任何可用介质。以此为例但不限于:计算机可读介质可以包括RAM、ROM、EEPROM、CD-ROM或其他光盘存储、磁盘存储介质或者其他磁存储设备、或者能够用于携带或存储具有指令或数据结构形式的期望的程序代码并能够由计算机存取的任何其他介质。此外,任何连接可以适当的成为计算机可读介质。例如,如果软件是使用同轴电缆、光纤光缆、双绞线、数字用户线(DSL)或者诸如红外线、无线电和微波之类的无线技术从网站、服务器或者其他远程源传输的,那么同轴电缆、光纤光缆、双绞线、DSL或者诸如红外线、无线和微波之类的无线技术包括在所属介质的定影中。如本申请所使用的,盘(Disk)和碟(disc)包括压缩光碟(CD)、激光碟、光碟、数字通用光碟(DVD)、软盘和蓝光光碟,其中盘通常磁性的复制数据,而碟则用激光来光学的复制数据。上面的组合也应当包括在计算机可读介质的保护范围之内。

[0298] 在上述实施例中,可以全部或部分地通过软件、硬件、固件或者其任意组合来实现。当使用软件实现时,可以全部或部分地以计算机程序产品的形式实现。

[0299] 所述计算机程序产品包括一个或多个计算机指令。在计算机上加载和执行所述计算机程序指令时,全部或部分地产生按照本申请实施例所述的流程或功能。所述计算机可以是通用计算机、专用计算机、计算机网络、或者其他可编程装置。所述计算机指令可以存储在计算机可读存储介质中,或者从一个计算机可读存储介质向另一计算机可读存储介质传输,例如,所述计算机指令可以从一个网站站点、计算机、服务器或数据中心通过有线(例

如同轴电缆、光纤、数字用户线(DSL)或无线(例如红外、无线、微波等)方式向另一个网站站点、计算机、服务器或数据中心进行传输。所述计算机可读存储介质可以是计算机能够存储的任何可用介质或者是包含一个或多个可用介质集成的服务器、数据中心等数据存储设备。所述可用介质可以是磁性介质,(例如,软盘、硬盘、磁带)、光介质(例如,DVD)、或者半导体介质(例如固态硬盘Solid State Disk(SSD))等。

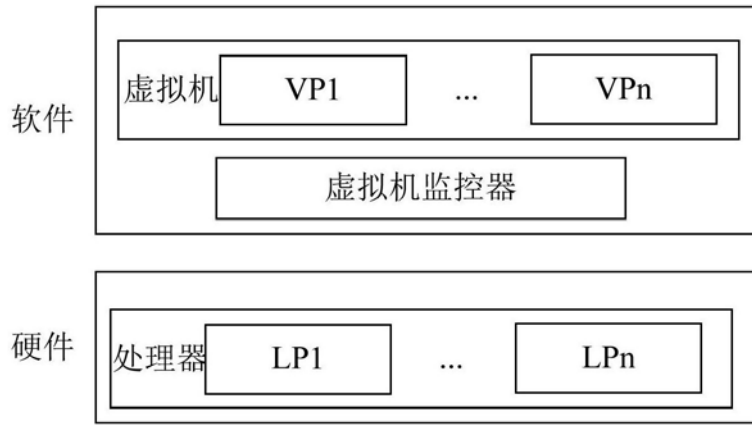


图1

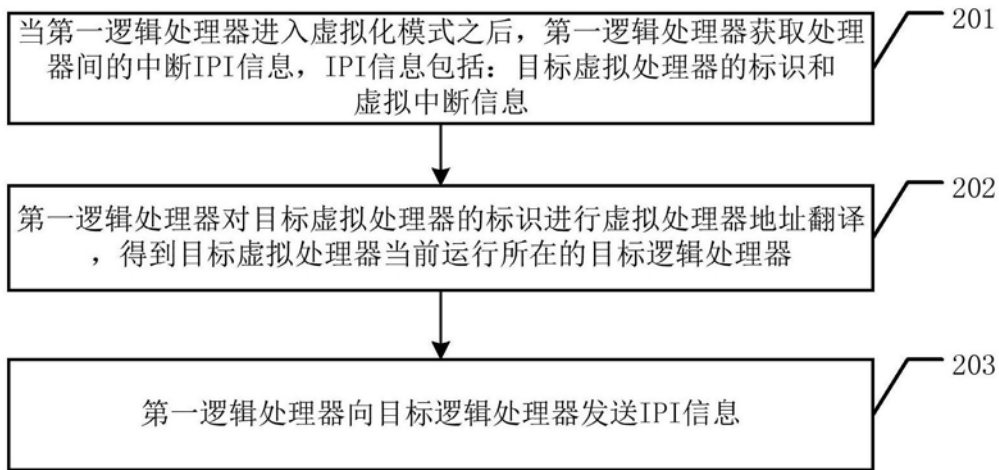


图2

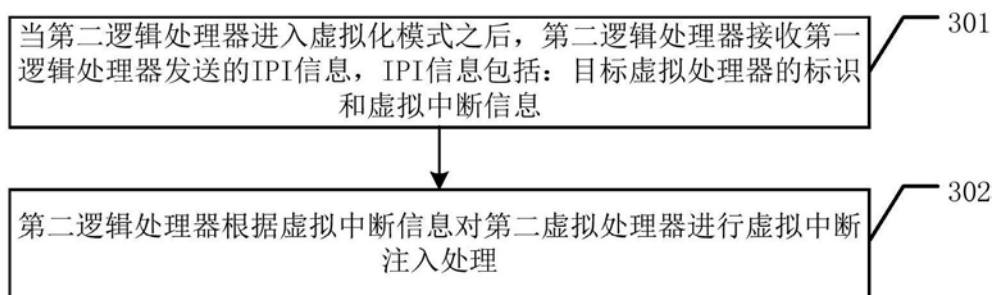


图3

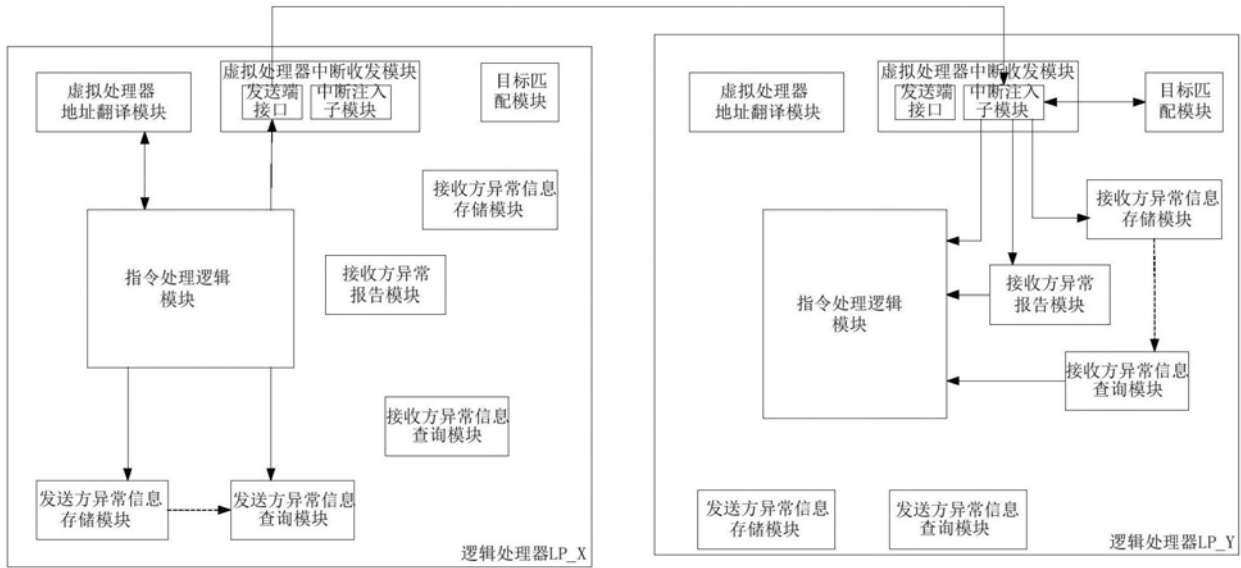


图4

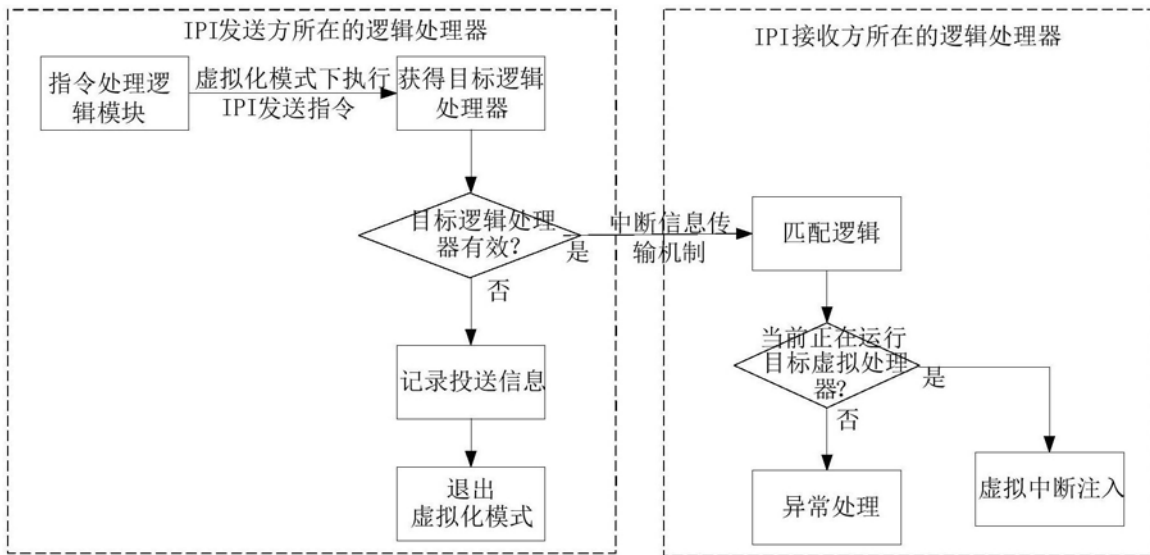


图5

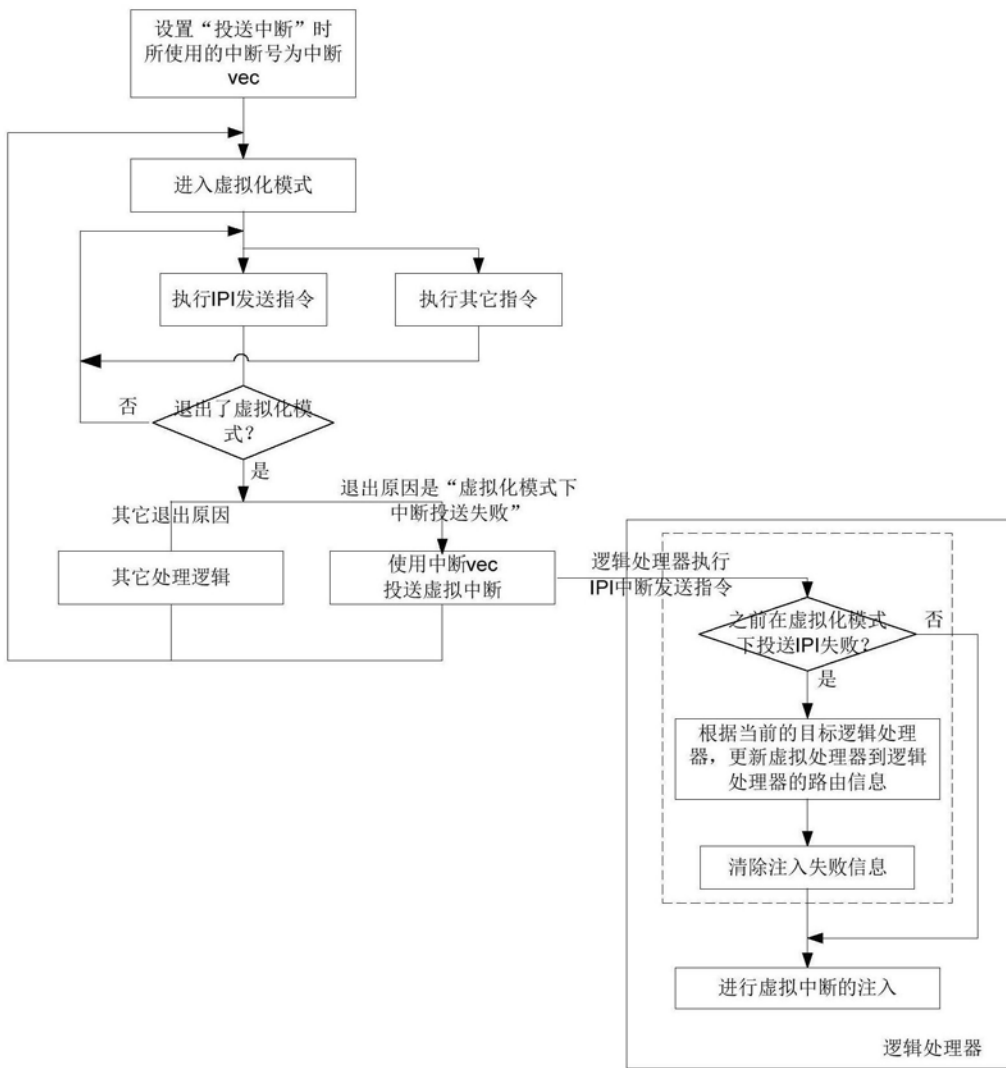


图6

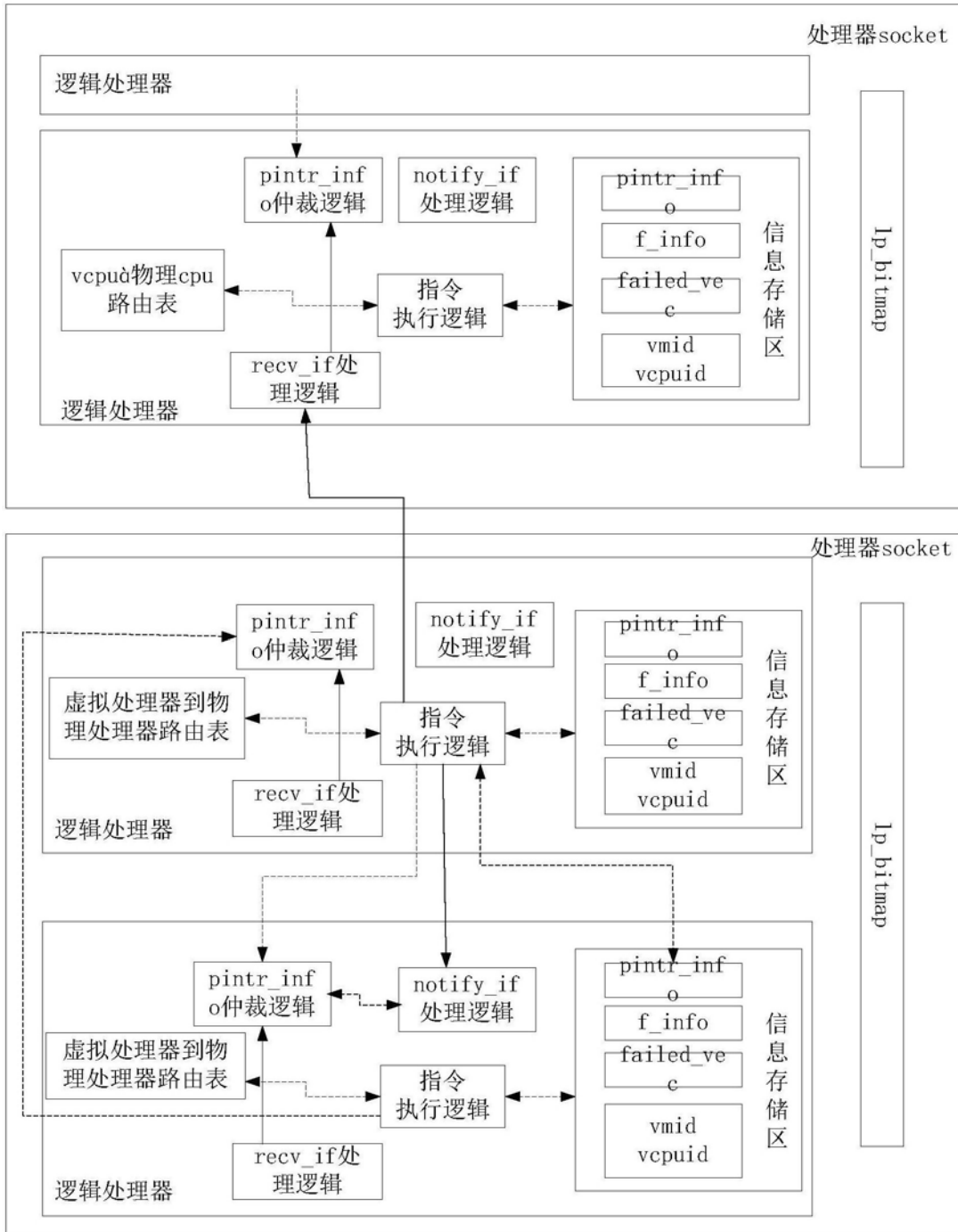


图7



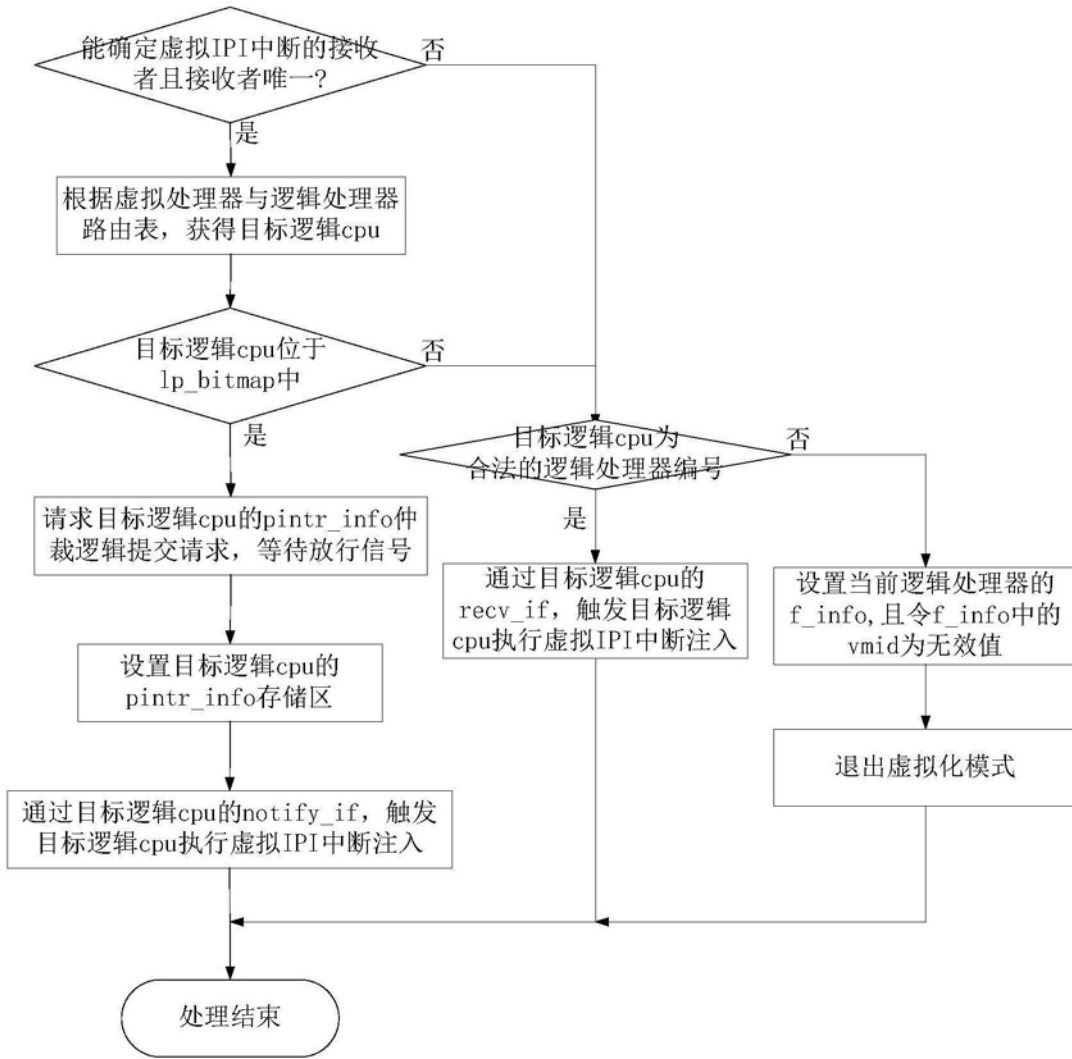


图8

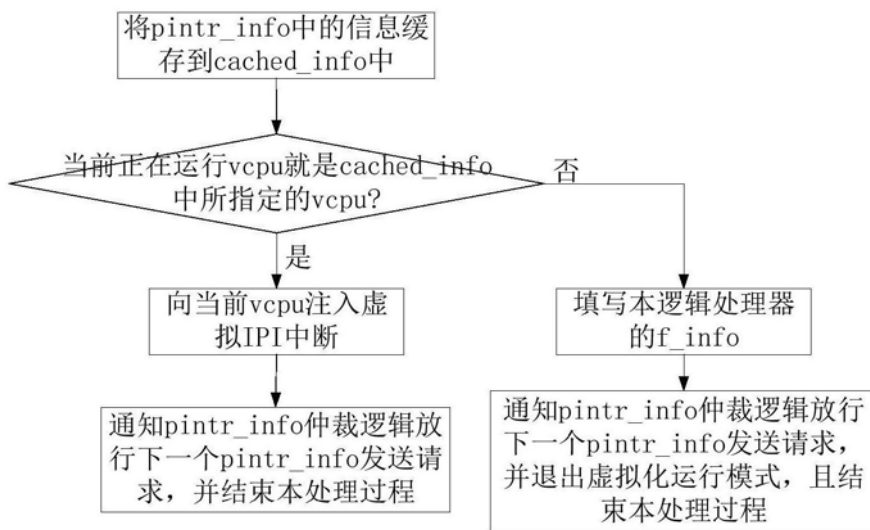


图9

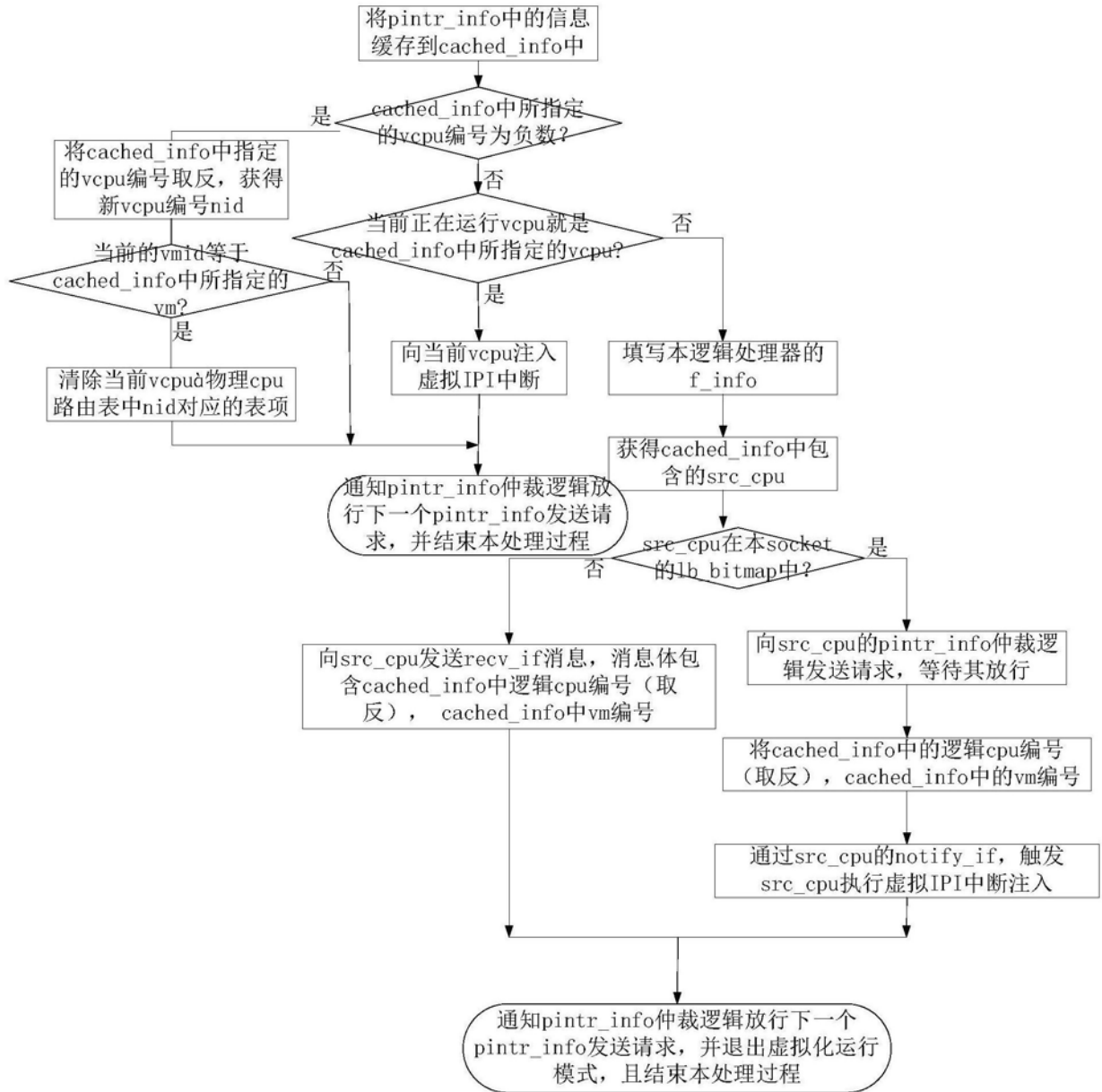


图10

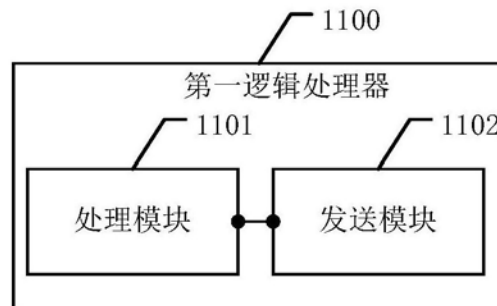


图11

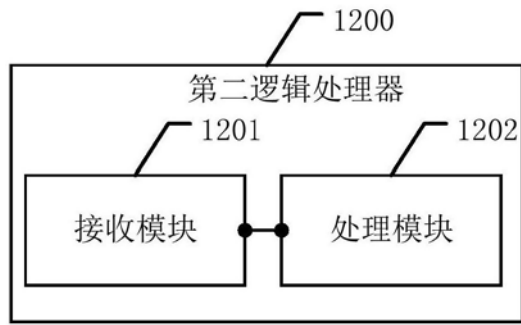


图12