(12) 按照专利合作条约所公布的国际申请

(19) 世界知识产权组织
国 际 局

(43) 国际公布日
2023 年 10 月 5 日 (05.10.2023)

WIPO | PCT

(10) 国际公布号
WO 2023/185478 A1

(51) 国际专利分类号:
G06F 9/54 (2006.01)

(21) 国际申请号: PCT/CN2023/081645

(22) 国际申请日: 2023 年 3 月 15 日 (15.03.2023)

(25) 申请语言: 中文

(26) 公布语言: 中文

(30) 优先权:
202210322866.9 2022年3月29日 (29.03.2022) CN

(71) 申请人: 华 为 技 术 有 限 公 司 (**HUAWEI TECHNOLOGIES CO., LTD.**) [CN/CN]; 中国广东省深圳市龙岗区坂田华为总部办公楼, Guangdong 518129 (CN)。

(72) 发明人: 白 卫 华 (**BAI, Weihua**); 中国广东省深圳市龙岗区坂田华为总部办公楼, Guangdong 518129 (CN)。 肖威(**XIAO, Wei**); 中国广东省深圳市龙岗区坂田华为总部办公楼, Guangdong 518129 (CN)。 柴先平 (**CHAI, Xianping**); 中国广东省深圳市龙岗区坂田华为总部办公楼, Guangdong 518129 (CN)。

(74) 代理人: 北 京 格 罗 巴 尔 知 识 产 权 代 理 事 务 所 (普 通 合 伙) (**BEIJING GLOBAL INTELLECTUAL PROPERTY LAW FIRM**); 中国北京市东城区北三环东路36号2号楼C1912房间, Beijing 100013 (CN)。

(81) 指定国(除另有指明，要求每一种可提供的国家保护): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CV, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, MG, MK, MN, MU, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA,

(54) **Title:** METHOD AND APPARATUS FOR COMMUNICATION BETWEEN APPLICATION PROGRAMS, AND STORAGE MEDIUM AND PROGRAM PRODUCT

(54) 发明名称: 应用程序间通信方法、装置、存储介质及程序产品

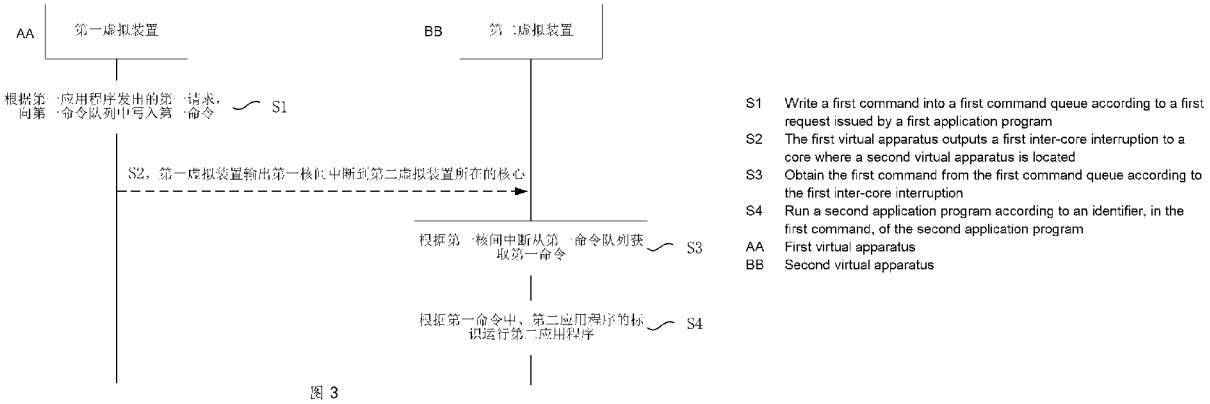


图 3

(57) **Abstract:** The present application relates to a method and apparatus for communication between application programs, and a storage medium and a program product. The method is applied to an electronic device, the electronic device runs a first virtual apparatus and a second virtual apparatus, one of the first virtual apparatus and the second virtual apparatus is a virtual machine, and the other one of the first virtual apparatus and the second virtual apparatus is a physical machine. The method comprises: a first virtual apparatus writes a first command into a first command queue according to a first request issued by a first application program; the first virtual apparatus outputs a first inter-core interruption to a core where a second virtual apparatus is located; the second virtual apparatus obtains the first command from the first command queue according to the first inter-core interruption; and the second virtual apparatus runs a second application program according to an identifier, in the first command, of the second application program. According to the method for communication between the application programs of the embodiments of the present application, fast communication between a service of the virtual machine and a service of the physical machine can be achieved, and the communication efficiency of the electronic device can be improved.

WO 2023/185478 A1

[见续页]

PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW。

(84) 指定国(除另有指明，要求每一种可提供的地区保护)：ARIPO (BW, CV, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SC, SD, SL, ST, SZ, TZ, UG, ZM, ZW), 欧亚 (AM, AZ, BY, KG, KZ, RU, TJ, TM), 欧洲 (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, ME, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG)。

本国际公布:

— 包括国际检索报告(条约第21条(3))。

(57) 摘要：本申请涉及一种应用程序间通信方法、装置、存储介质及程序产品，所述方法应用于电子设备，电子设备运行第一虚拟装置和第二虚拟装置，第一虚拟装置和第二虚拟装置中的一个是虚拟机，另一个是物理机，该方法包括：第一虚拟装置根据第一应用程序发出的第一请求，向第一命令队列中写入第一命令；第一虚拟装置输出第一核间中断到第二虚拟装置所在的核心；第二虚拟装置根据第一核间中断从第一命令队列获取第一命令；第二虚拟装置根据第一命令中、第二应用程序的标识运行第二应用程序。根据本申请实施例的应用程序间通信方法，能够实现虚拟机的业务和物理机的业务之间的快速通信，提高电子设备的通信效率。

应用程序间通信方法、装置、存储介质及程序产品

本申请要求于2022年3月29日提交中国专利局、申请号为202210322866.9、申请名称为“应用程序间通信方法、装置、存储介质及程序产品”的中国专利申请的优先权，其全部内容通过引用结合在本申请中。

技术领域

本申请涉及通信领域，尤其涉及一种应用程序间通信方法、装置、存储介质及程序产品。

背景技术

随着软硬件的发展和应用场景的变化，对于支持多操作系统同时部署的嵌入式系统的需求越来越强。以智能汽车为例，同一款芯片需运行多种完全不同的业务，这些业务对于计算资源、执行环境的需求是完全不同的。如有些和人身安全强相关的业务需要操作系统提供强实时性，而音乐媒体业务则需要操作系统具有较高的生态兼容性，对操作系统的实时性要求很低。这些不同的需求的达成，需要嵌入式系统的操作系统的密切配合。

针对于不同的业务需求，嵌入式系统通常使用虚拟化技术部署多个不同类型的操作系统，如实时操作系统、控制操作系统、安全操作系统等。在物理机上，通过一个虚拟机监控器同时管理运行上述多个操作系统，即运行多个虚拟机，并根据需求将不同种类的业务运行在不同的虚拟机中，其中物理机也称为宿主机，被物理机运行的多个虚拟机也可称为客户机。

按照通用虚拟化的特点，不同的虚拟机和物理机之间是互相隔离运行的。但是针对于某些特定的场景或者需求，虚拟机的业务和物理机的业务可能需要相互通信。

发明内容

有鉴于此，提出了一种应用程序间通信方法、装置、存储介质及程序产品，根据本申请实施例的应用程序间通信方法，能够实现虚拟机的业务和物理机的业务之间的快速通信，提高电子设备的通信效率。

第一方面，本申请的实施例提供了一种应用程序间通信方法，所述方法应用于电子设备，所述电子设备包括第一虚拟装置和第二虚拟装置，所述第一虚拟装置和所述第二虚拟装置中的一个是虚拟机，另一个是物理机，所述方法包括：所述第一虚拟装置根据第一应用程序发出的第一请求，向第一命令队列中写入第一命令，所述第一请求、所述第一命令包括所述第二虚拟装置上的第二应用程序的标识；所述第一虚拟装置输出第一核间中断到所述第二虚拟装置所在的核心，以使所述第二虚拟装置能够获取所述第一核间中断；所述第二虚拟装置根据所述第一核间中断从所述第一命令队列获取所述第一命令；所述第二虚拟装置根据所述第一命令中、所述第二应用程序的标识运行所述第二应用程序。

根据本申请实施例的应用程序间通信方法，通过第一虚拟装置根据第一应用程序发出的第一请求向第一命令队列中写入第一命令，并输出第一核间中断到第二虚拟装置所在的核心，使得第二虚拟装置可以获取第一核间中断并根据第一核间中断从第一命令队列获取第一命令，

可以实现第一虚拟装置到第二虚拟装置的通信；第一命令是根据来自第一应用程序的第一请求写入的，并包括第二虚拟装置上的第二应用程序的标识，使得第二虚拟装置根据第一命令中、第二应用程序的标识可以运行第二应用程序，从而能够实现第一虚拟装置的应用程序到第二虚拟装置的应用程序的通信。由于第一虚拟装置和所述第二虚拟装置中的一个是虚拟机，另一个是物理机，因此物理机和虚拟机之间使用将核间中断直接输出至需通信的对象所在的核心的方式完成通信，可以避免使用虚拟中断控制器，从而避免虚拟机的陷入陷出，可以实现虚拟机的业务和物理机的业务之间的快速通信，提高电子设备的通信效率。

根据第一方面，在所述应用程序间通信方法的第一种可能的实现方式中，所述第一虚拟装置还包括第一代理模块，所述第一虚拟装置根据第一应用程序发出的第一请求，向第一命令队列中写入第一命令，包括：所述第一代理模块接收所述第一应用程序发送的所述第一请求；所述第一代理模块根据所述第一请求，向所述第一命令队列中写入所述第一命令；所述第一虚拟装置输出第一核间中断到所述第二虚拟装置所在的核心，包括：所述第一代理模块输出所述第一核间中断到所述第二虚拟装置所在的核心。

通过这种方式，可以使用第一代理模块完成与第一应用程序相关的部分工作，避免第一应用程序直接与第二虚拟装置本身进行数据交换，保证了第一应用程序运行的安全性，同时降低了第一应用程序的数据处理成本。

根据第一方面或第一方面的第一种可能的实现方式，在所述应用程序间通信方法的第二种可能的实现方式中，所述第二虚拟装置还包括第二代理模块，所述第二虚拟装置根据所述第一核间中断从所述第一命令队列获取所述第一命令包括：所述第二代理模块根据所述第一核间中断从所述第一命令队列获取所述第一命令；所述第二虚拟装置根据所述第一命令中、所述第二应用程序的标识运行所述第二应用程序包括：所述第二代理模块根据所述第一命令中、所述第二应用程序的标识运行所述第二应用程序。

通过这种方式，可以使用第二代理模块完成与第二应用程序相关的部分工作，避免第二应用程序直接与第一虚拟装置本身进行数据交换，保证了第二应用程序运行的安全性，同时降低了第二应用程序的数据处理成本。

根据第一方面的第一种或第二种可能的实现方式，在所述应用程序间通信方法的第三种可能的实现方式中，所述第一请求和所述第一命令还包括所述第一应用程序的阻塞信息，所述第一代理模块接收所述第一应用程序发送的所述第一请求之后，所述方法还包括：所述第一应用程序进入阻塞状态；在接收到第二核间中断时，所述第一代理模块根据所述第二核间中断从第二命令队列获取第二命令，所述第二核间中断在所述第二应用程序运行结束后产生；在所述第二命令包括所述第一应用程序的标识时，所述第一代理模块解除所述第一应用程序的阻塞状态。

通过这种方式，对于发出向第二应用程序通信的请求后阻塞的第一应用程序，可以实现第二应用程序到第一应用程序的快速通知，阻塞的第一应用程序可以通过该通知解除阻塞状态，使得可以保证第一应用程序可以快速地继续运行。

根据第一方面的第三种可能的实现方式，在所述应用程序间通信方法的第四种可能的实现方式中，所述第二虚拟装置根据所述第一命令中、所述第二应用程序的标识运行所述第二应用程序之后，所述方法还包括：所述第二应用程序运行结束后，所述第二代理模块接收所述第二应用程序发送的第二请求，所述第二请求包括所述第一应用程序的标识；所述第二代

理模块根据所述第二请求，向所述第二命令队列中写入所述第二命令，所述第二命令包括所述第一应用程序的标识；所述第二代理模块输出所述第二核间中断到所述第一虚拟装置所在的核心，以使所述第一虚拟装置能够获取所述第二核间中断。

通过这种方式，使得第二代理模块可以确定第二代理模块运行结束后需通知的是第一应用程序，使得第一代理模块根据第二命令可以解除的是第一应用程序的阻塞状态，以保证第一应用程序解除阻塞状态时，第二应用程序确实已经运行结束，从而能满足第一应用程序等待第二应用程序运行完毕后继续运行的需求。

根据第一方面的第一种至第四种可能的实现方式中的任意一种可能的实现方式，在所述应用程序间通信方法的第五种可能的实现方式中，所述第一代理模块根据所述第一请求，向所述第一命令队列中写入所述第一命令，包括：所述第一代理模块根据所述第一请求，判断所述第二应用程序是否具有通知权限；在所述第二应用程序具有通知权限时，向所述第一命令队列中写入所述第一命令。

通过这种方式，可以保证接收到通知的应用程序是具有通知权限的应用程序，保证通知的安全性。

根据第一方面的第三种至第五种可能的实现方式中的任意一种可能的实现方式，在所述应用程序间通信方法的第六种可能的实现方式中，所述第一虚拟装置根据所述第一应用程序发出的第一请求，向第一命令队列中写入第一命令之前，所述方法还包括：响应于所述第一应用程序的申请，所述第一虚拟装置创建共享内存，所述第一代理模块建立所述第一应用程序的地址与所述共享内存的地址的关联关系；所述第一请求和所述第一命令还包括所述共享内存的参数；在所述第二命令包括所述第一应用程序的标识时，所述第一代理模块解除所述第一应用程序的阻塞状态，包括：在所述第二命令包括所述第一应用程序的标识以及通道建立成功的信息时，所述第一代理模块解除所述第一应用程序的阻塞状态。

通过这种方式，使得可以通过共享内存作为第一应用程序和第二应用程序之间的专用数据通道，相比仅采用命令队列的形式进行数据传递的方案，可以传递更丰富的数据。同时共享内存可传递的数据量比较大，可以减少数据的拷贝次数，提升数据的传输效率。

根据第一方面的第六种可能的实现方式，在所述应用程序间通信方法的第七种可能的实现方式中，所述第二代理模块根据所述第一命令中、所述第二应用程序的标识运行所述第二应用程序，包括：所述第二代理模块根据所述第一命令，判断所述第二应用程序是否具有访问所述共享内存的权限；所述第二代理模块运行所述第二应用程序，并在所述第二应用程序具有访问所述共享内存的权限时，建立所述第二应用程序的地址与所述共享内存的地址的关联关系；所述第二命令还包括通道建立成功或通道建立失败的信息。

通过这种方式，使得专用数据通道建立前，第二代理模块可以确定第二应用程序是否具备访问共享内存的权限，专用数据通道在第二应用程序具备访问共享内存的权限时建立，以保证专用数据通道的建立的安全性。通过第二命令将通道建立成功或通道建立失败的信息传递给第一虚拟装置侧，以使第一虚拟装置的第一应用程序可以确定其请求的执行情况。

根据第一方面的第六种或第七种可能的实现方式，在所述应用程序间通信方法的第八种可能的实现方式中，所述共享内存还作为控制通道，所述控制通道包括第一数据位和第二数据位，所述第一数据位和所述第二数据位的状态由所述第一代理模块和所述第二代理模块控制；所述第一虚拟装置输出第一核间中断到所述第二虚拟装置所在的核心之前，所述方法还

包括：所述第一代理模块控制所述第一数据位处于置位状态、第二数据位处于非置位状态；所述第一代理模块输出所述第一核间中断到所述第二虚拟装置所在的核心，包括：所述第一代理模块确定所述第一应用程序和所述第二应用程序之间的控制通道的第一数据位处于置位状态、第二数据位处于非置位状态时，所述第一代理模块输出第一核间中断到所述第二虚拟装置所在的核心。

通过控制通道专用于控制核间中断的发送及接收，可以降低核间中断的收发次数，降低应用程序间通信方法的实现复杂度。

根据第一方面的第八种可能的实现方式，在所述应用程序间通信方法的第九种可能的实现方式中，所述第二代理模块根据所述第一核间中断从所述第一命令队列获取所述第一命令，包括：所述第二代理模块根据所述第一核间中断，遍历所述第二虚拟装置上、所述第二代理模块对应的全部应用程序关联的控制通道；在确定所述第一应用程序和所述第二应用程序之间的控制通道的第一数据位处于置位状态时，所述第二代理模块从第一命令队列获取所述第一命令。

通过第二代理模块对于其对应的全部应用程序关联的控制通道的第一数据位均进行检查，可以确定是哪一通道对应的应用程序有请求发出；通过第二代理模块根据第一数据位的状态确定是否需获取命令，可以避免命令的重复获取，可以提高第二代理模块的工作效率。

根据第一方面的第八种或第九种可能的实现方式，在所述应用程序间通信方法的第十种可能的实现方式中，所述第二虚拟装置根据所述第一命令中、所述第二应用程序的标识运行所述第二应用程序之后，所述方法还包括：所述第二代理模块控制所述第一应用程序和所述第二应用程序之间的控制通道的第一数据位处于非置位状态、第二数据位处于非置位状态；针对所述第一应用程序和所述第二应用程序之间的控制通道，所述第二应用程序运行结束后，所述第二代理模块确定该控制通道的第一数据位处于非置位状态时，控制该控制通道的第二数据位处于置位状态；所述第二代理模块确定该控制通道的第一数据位处于置位状态时，重新从所述第一命令队列获取命令，并根据新获取的命令中第二应用程序的标识调度所述第二应用程序运行。

通过这种方式，可以在第一应用程序和所述第二应用程序之间的控制通道的请求处理完毕后，继续处理第一应用程序后续发起的新请求，而不必先获取核间中断后再对新请求进行处理，可以降低核间中断的收发次数，节约数据传输成本。

根据第一方面的第五种至第十种可能的实现方式中的任意一种可能的实现方式，在所述应用程序间通信方法的第十一种可能的实现方式中，在所述第一虚拟装置是物理机、所述第二虚拟装置是虚拟机时，所述第一虚拟装置还包括鉴权模块和第一进程模块，所述第二代理模块根据所述第一命令，判断所述第二应用程序是否具有访问所述共享内存的权限，包括：所述第二代理模块将所述第一命令中、所述第二应用程序的标识转发至所述第一虚拟装置上的第一进程模块；所述第一进程模块确定与所述第二应用程序的标识对应的鉴权标识，并向所述鉴权模块发送权限信息获取请求，所述权限信息获取请求包括所述鉴权标识；所述第一进程模块接收来自所述鉴权模块的所述第二应用程序的权限信息，并传输至所述第二代理模块；所述第二代理模块根据所述第二应用程序的权限信息，判断所述第二应用程序是否具有访问所述共享内存的权限。

在第一虚拟装置是物理机时，第一进程模块可作为虚拟机侧和物理机侧的接口，使得虚

拟机侧的应用程序的相关信息可以通过第一进程模块传输至物理机侧，从而可以利用物理机侧的鉴权模块为虚拟机侧的应用程序进行鉴权。通过鉴权模块进行鉴权，可以进一步提高数据传输的安全性。

第二方面，本申请的实施例提供了一种应用程序间通信装置，所述装置应用于电子设备，所述电子设备运行第一虚拟装置和第二虚拟装置，所述第一虚拟装置和所述第二虚拟装置中的一个是虚拟机，另一个是物理机，所述装置包括：所述第一虚拟装置，用于根据第一应用程序发出的第一请求，向第一命令队列中写入第一命令，所述第一请求、所述第一命令包括第二虚拟装置上的第二应用程序的标识；输出第一核间中断到所述第二虚拟装置所在的核心，以使所述第二虚拟装置能够获取所述第一核间中断；所述第二虚拟装置，用于根据所述第一核间中断从所述第一命令队列获取所述第一命令；根据所述第一命令中、所述第二应用程序的标识运行所述第二应用程序。

根据第二方面，在所述应用程序间通信装置的第一种可能的实现方式中，所述第一虚拟装置还包括第一代理模块，所述第一代理模块用于，接收所述第一应用程序发送的所述第一请求；根据所述第一请求，向所述第一命令队列中写入所述第一命令；输出所述第一核间中断到所述第二虚拟装置所在的核心。

根据第二方面或第二方面的第一种可能的实现方式，在所述应用程序间通信装置的第二种可能的实现方式中，所述第二虚拟装置还包括第二代理模块，所述第二代理模块用于，根据所述第一核间中断从所述第一命令队列获取所述第一命令；根据所述第一命令中、所述第二应用程序的标识运行所述第二应用程序。

根据第二方面的第一种或第二种可能的实现方式，在所述应用程序间通信装置的第三种可能的实现方式中，所述第一请求和所述第一命令还包括所述第一应用程序的阻塞信息，所述第一代理模块接收所述第一应用程序发送的所述第一请求之后，所述第一应用程序进入阻塞状态；在接收到第二核间中断时，所述第一代理模块还用于，根据所述第二核间中断从第二命令队列获取第二命令，所述第二核间中断在所述第二应用程序运行结束后产生；在所述第二命令包括所述第一应用程序的标识时，所述第一代理模块还用于，解除所述第一应用程序的阻塞状态。

根据第二方面的第三种可能的实现方式，在所述应用程序间通信装置的第四种可能的实现方式中，所述第二应用程序运行结束后，所述第二代理模块还用于，接收所述第二应用程序发送的第二请求，所述第二请求包括所述第一应用程序的标识；根据所述第二请求，向所述第二命令队列中写入所述第二命令，所述第二命令包括所述第一应用程序的标识；输出所述第二核间中断到所述第一虚拟装置所在的核心，以使所述第一虚拟装置能够获取所述第二核间中断。

根据第二方面的第一种至第四种可能的实现方式中的任意一种可能的实现方式，在所述应用程序间通信装置的第五种可能的实现方式中，所述第一代理模块根据所述第一请求，向所述第一命令队列中写入所述第一命令，包括：所述第一代理模块根据所述第一请求，判断所述第二应用程序是否具有通知权限；在所述第二应用程序具有通知权限时，向所述第一命令队列中写入所述第一命令。

根据第二方面的第三种至第五种可能的实现方式中的任意一种可能的实现方式，在所述应用程序间通信装置的第六种可能的实现方式中，所述第一虚拟装置还用于响应于所述第一

应用程序的申请创建共享内存，所述第一代理模块还用于，建立所述第一应用程序的地址与所述共享内存的地址的关联关系；所述第一请求和所述第一命令还包括所述共享内存的参数；在所述第二命令包括所述第一应用程序的标识时，所述第一代理模块解除所述第一应用程序的阻塞状态，包括：在所述第二命令包括所述第一应用程序的标识以及通道建立成功的信息时，所述第一代理模块解除所述第一应用程序的阻塞状态。

根据第二方面的第六种可能的实现方式，在所述应用程序间通信装置的第七种可能的实现方式中，所述第二代理模块根据所述第一命令中、所述第二应用程序的标识运行所述第二应用程序，包括：所述第二代理模块根据所述第一命令，判断所述第二应用程序是否具有访问所述共享内存的权限；所述第二代理模块运行所述第二应用程序，并在所述第二应用程序具有访问所述共享内存的权限时，建立所述第二应用程序的地址与所述共享内存的地址的关联关系；所述第二命令还包括通道建立成功或通道建立失败的信息。

根据第二方面的第六种或第七种可能的实现方式，在所述应用程序间通信装置的第八种可能的实现方式中，所述共享内存还作为控制通道，所述控制通道包括第一数据位和第二数据位，所述第一数据位和所述第二数据位的状态由所述第一代理模块和所述第二代理模块控制；所述第一代理模块还用于，控制所述第一数据位处于置位状态、第二数据位处于非置位状态；所述第一代理模块输出所述第一核间中断到所述第二虚拟装置所在的核心，包括：所述第一代理模块确定所述第一应用程序和所述第二应用程序之间的控制通道的第一数据位处于置位状态、第二数据位处于非置位状态时，所述第一代理模块输出第一核间中断到所述第二虚拟装置所在的核心。

根据第二方面的第八种可能的实现方式，在所述应用程序间通信装置的第九种可能的实现方式中，所述第二代理模块根据所述第一核间中断从所述第一命令队列获取所述第一命令，包括：所述第二代理模块根据所述第一核间中断，遍历所述第二虚拟装置上、所述第二代理模块对应的全部应用程序关联的控制通道；在确定所述第一应用程序和所述第二应用程序之间的控制通道的第一数据位处于置位状态时，所述第二代理模块从第一命令队列获取所述第一命令。

根据第二方面的第八种或第九种可能的实现方式，在所述应用程序间通信装置的第十种可能的实现方式中，所述第二代理模块还用于，控制所述第一应用程序和所述第二应用程序之间的控制通道的第一数据位处于非置位状态、第二数据位处于非置位状态；针对所述第一应用程序和所述第二应用程序之间的控制通道，所述第二应用程序运行结束后，所述第二代理模块还用于，确定该控制通道的第一数据位处于非置位状态时，控制该控制通道的第二数据位处于置位状态；所述第二代理模块还用于，确定该控制通道的第一数据位处于置位状态时，重新从所述第一命令队列获取命令，并根据新获取的命令中第二应用程序的标识调度所述第二应用程序运行。

根据第二方面的第五种至第十种可能的实现方式中的任意一种可能的实现方式，在所述应用程序间通信装置的第十一种可能的实现方式中，在所述第一虚拟装置是物理机、所述第二虚拟装置是虚拟机时，所述第一虚拟装置还包括鉴权模块和第一进程模块，所述第二代理模块根据所述第一命令，判断所述第二应用程序是否具有访问所述共享内存的权限，包括：所述第二代理模块将所述第一命令中、所述第二应用程序的标识转发至所述第一虚拟装置上的第一进程模块；所述第一进程模块确定与所述第二应用程序的标识对应的鉴权标识，并向

所述鉴权模块发送权限信息获取请求，所述权限信息获取请求包括所述鉴权标识；所述第一进程模块接收来自所述鉴权模块的所述第二应用程序的权限信息，并传输至所述第二代理模块；所述第二代理模块根据所述第二应用程序的权限信息，判断所述第二应用程序是否具有访问所述共享内存的权限。

第三方面，本申请的实施例提供了一种应用程序间通信装置，包括：处理器；用于存储处理器可执行指令的存储器；其中，所述处理器被配置为执行所述指令时实现上述第一方面或者第一方面的多种可能的实现方式中的一种或几种的应用程序间通信方法。

第四方面，本申请的实施例提供了一种非易失性计算机可读存储介质，其上存储有计算机程序指令，所述计算机程序指令被处理器执行时实现上述第一方面或者第一方面的多种可能的实现方式中的一种或几种的应用程序间通信方法。

第五方面，本申请的实施例提供了一种计算机程序产品，包括计算机可读代码，或者承载有计算机可读代码的非易失性计算机可读存储介质，当所述计算机可读代码在电子设备中运行时，所述电子设备中的处理器执行上述第一方面或者第一方面的多种可能的实现方式中的一种或几种的应用程序间通信方法。

本申请的这些和其他方面在以下（多个）实施例的描述中会更加简明易懂。

附图说明

包含在说明书中并且构成说明书的一部分的附图与说明书一起示出了本申请的示例性实施例、特征和方面，并且用于解释本申请的原理。

图 1 示出现有技术二的基于网络 posix 接口的相互通信方案的工作流程示意图。

图 2 示出根据本申请实施例的应用程序间通信方法的示例性应用场景。

图 3 示出根据本申请实施例的应用程序间通信方法的示例性流程示意图。

图 4 示出根据本申请实施例的第一应用程序通知第二应用程序的示例性方法的示意图。

图 5a 示出根据本申请实施例的第一虚拟装置作为实体接收核间中断的一个示例。

图 5b 示出根据本申请实施例的第一虚拟装置作为实体接收核间中断的另一个示例。

图 6 示出根据本申请实施例的第一应用程序通知第二应用程序之后接收到第二应用程序的通知的示例性方法的示意图。

图 7 示出根据本申请实施例建立专用数据通道的示例性示意图。

图 8 示出根据本申请实施例的控制通道的一个示例。

图 9 示出根据本申请实施例的第一应用程序和所述第二应用程序之间的控制通道、专用数据通道、标识和信号量进行绑定的示例。

图 10a 示出根据本申请实施例确定第二应用程序是否具备访问所述共享内存的权限的示例性方法示意图。

图 10b 示出根据本申请实施例确定第二应用程序是否具备访问所述共享内存的权限的示例性方法示意图。

图 11 示出根据本申请实施例的应用程序间通信方法的一个具体示例。

图 12 示出根据本申请实施例的应用程序间通信装置 10 的示例性结构示意图。

图 13 示出根据本申请实施例的应用程序间通信装置的示例性结构示意图。

具体实施方式

以下将参考附图详细说明本申请的各种示例性实施例、特征和方面。附图中相同的附图标记表示功能相同或相似的元件。尽管在附图中示出了实施例的各种方面，但是除非特别指出，不必按比例绘制附图。

在这里专用的词“示例性”意为“用作例子、实施例或说明性”。这里作为“示例性”所说明的任何实施例不必解释为优于或好于其它实施例。

另外，为了更好的说明本申请，在下文的具体实施方式中给出了众多的具体细节。本领域技术人员应当理解，没有某些具体细节，本申请同样可以实施。在一些实例中，对于本领域技术人员熟知的方法、手段、元件和电路未作详细描述，以便于凸显本申请的主旨。

以下对本文中出现的术语进行解释。

虚拟化（Virtualization）：虚拟化是指将真实计算机元器件、计算资源进行抽象，并重新分配，建立一个或者多个虚拟环境，让计算元件在此虚拟的环境中运行的技术。抽象的对象包括处理器、内存、输入/输出接口外设等。虚拟化技术可以简化软件的重新配置过程，让一个平台可同时运行多个实例，并让所有的系统和应用程序都运行于互不干扰的不同空间。

虚拟机（virtual machine，VM）：在虚拟化提供的某个虚拟环境中运行的实例就是一个虚拟机。

实时性：任务在某个确定性时延内得到执行的能力。

嵌入式系统（embedded system）：嵌入式系统由硬件和软件组成，是一种嵌入在非通用计算机的电子/机械器件中的操作系统，通常具有小型化、高实时性的特点。

虚拟机监视器（virtual machine monitor，VMM）：用来创建与运行虚拟机的软件、固件或硬件，例如hypervisor。

宿主机（host machine）：又称host OS，简称host，被虚拟机监视器用来运行一个或多个虚拟机的系统或机器称为宿主机。

客户机（guest machine）：又称guest OS，简称guest，被宿主机运行的虚拟机称为客户机。

应用程序（app）：即宿主机或客户机运行的业务，可以理解为一个有特定功能的程序。

虚拟机陷出（trap）：虚拟机通过硬件能力从虚拟环境中切换到物理环境中运行，可以因异常、中断等被动陷出，也可通过指令主动陷出。

内核态：处理器切换到特权模式，可以访问所有硬件资源，包括内存、输出/输出接口、计算资源等。内核态负责管理应用程序对于上述硬件资源的分配和使用。

虚拟中断控制器：用于接收多个外部中断源的中断请求，并进行优先级判断，选中当前优先级最高的中断请求，并将此请求送给处理器。

阻塞：应用程序因为某种资源暂时不可得而处于停止运行状态，等待资源满足要求后可解除阻塞状态。

特权级：现代处理器在运行时，软件可能会运行在不同的特权级，高特权级通常比低特权级有更多权限。

驱动:运行在操作系统中的一种软件(一般运行于较高特权级别),用来操作外设的行为。

前后端驱动：在虚拟化场景中，因为真实的硬件只能在物理环境中进行管理，所以虚拟

机只能看到虚拟出来的硬件。在虚拟机中的虚拟外设的驱动便是前端驱动，在物理环境中负责模拟虚拟外设，并去请求真正外设进行操作的软件一般被叫做后端驱动。

信号量（futex）：本质上是一个原子变量，用来在多个对象之间进行同步，常用的方式是一方在信号量上阻塞，另一方通过给此信号量的变量+1 的方式，唤醒阻塞的一方。

2 阶段页表（stage2 页表）：因为虚拟环境是物理环境虚拟出来的，所以虚拟机看到的物理地址并不是真实的物理地址，需要一层页表翻译，转换为真实的物理地址。这个页表由软件维护，硬件自动翻译。此页表在一些平台上被称为 2 阶段页表。

超级调用（hypercall）：虚拟机主动通过一条超级调用指令请求物理环境中的服务，可以携带参数。

通信的含义包含两层，对于需通信的双方，第一层含义是通知对方，例如，一方可以通过某种机制阻塞，等待另一方通过某种方式与对方通信提醒一方继续运行；第二层含义则是消息内容的传递，即双方均可以通过某种通道，在通知对方的同时，给对方传递一些数据。这些数据可能是给对方的请求或者对于请求的答复，也可能是基于某种协议的数据。

在虚拟机的场景中，通用的虚拟化方案中，对于某一虚拟机和其外界的其他装置（例如其他虚拟机或物理机）是强调隔离性的，并不强调底层虚拟机和外界的通信行为。在不得不进行信息传递的时候，一般只能通过虚拟机陷出的方式，把消息带出来，然后再返回继续运行。虚拟机陷出和返回过程速度很慢，而且信息传递效率很低。所以在遇到性能瓶颈的时候，就需要有一种足够快速，又足够安全的机制，让位于物理机侧的应用程序和虚拟机侧的应用程序能够快速地相互通信。

现有技术主流使用以下两种方案实现物理机（宿主机）侧的应用程序和虚拟机（客户机）侧的应用程序之间的通信。

现有技术一提出了一种同步的共享内存机制，该方案的设计起源是为了解决如何使虚拟机中的驱动通过物理机使用真实外设的问题。该方案的实现方式如下：

当客户机想要通知宿主机时，客户机中的数据会通过共享内存的方式传递给宿主机，并将共享内存所在的地址通过 virtio 传输协议告知宿主机。当宿主机想要通知客户机侧时，一般通过中断注入的方式，当客户机被调度的时候，中断便会在虚拟机中上报，让客户机有机会处理宿主机侧的请求。

virtio 协议本身是为了解决虚拟机的前后端驱动间通信的问题，并非虚拟机和物理机内应用程序间通信的机制，现有技术一基于此协议进行改良，使得此机制可以用于客户机和宿主机的应用程序间通信。

当宿主机侧的应用程序想要通知客户机侧应用程序时，可共享内存中注明要通知的客户机侧的应用程序的信息，并迫使对应的虚拟机陷出，再向虚拟机中注入中断。而当客户机侧的应用程序想要通知宿主机侧的应用程序时，也可在共享内存中注明要通知的宿主机侧的应用程序的信息，并主动进行虚拟机陷出，再通过核间中断等手段通知给宿主机侧。

现有技术一的缺点在于，通过 virtio 协议进行通信时，势必会造成虚拟机的陷出，在 arm64 平台的物理机上，无论是宿主机向客户机还是客户机向宿主机的通知，都涉及虚拟机陷出。虚拟机的陷入陷出以及后续的保存上下文工作是很慢的流程，势必会造成性能瓶颈。并且，virtio 协议本身并没有提供任何安全机制，宿主机侧和客户机侧的应用程序通过 virtio 协议进行通信的时候，由于没有认证身份的流程，共享内存的访问向所有人开放，可

能会造成安全问题。

现有技术二提出了基于网络 posix 接口的相互通信方案。图 1 示出现有技术二的基于网络 posix 接口的相互通信方案的工作流程示意图。

如图 1 所示，该方案的原理是，只要宿主机为虚拟机提供了网络，则虚拟机中的应用程序也可以通过网络编程接口连接网络，并像普通应用程序一样使用网络。所以如果宿主机侧的应用程序和客户机侧的应用程序都通过如 posix 网络编程接口等方式使用网络，则可互相通信。为了加速客户机使用网络的性能，可以使用网卡直通的方式，该方式产生的网卡中断可以直接注入给虚拟机，让客户机使用网卡时无需陷入陷出。

现有技术二虽然免于虚拟机的陷入陷出，但是同样存在一些问题。其一，网络协议的数据传输效率是很低的，会造成数据的多次拷贝，如果通信传输的数据量大，会使通信时延大大增加。其二，即使使用网卡直通的方式加速网络性能，但由于网卡直通要依托于较苛刻的硬件环境，因此使用网卡直通意味着硬件成本的提高。对于无法直通的场景，或者没有可直通网卡的场景，网卡的使用同样会造成虚拟机的陷入陷出，降低性能。

有鉴于此，提出了一种应用程序间通信方法、装置、存储介质及程序产品，根据本申请实施例的应用程序间通信方法，能够实现虚拟机的业务和物理机的业务之间的快速通信，提高电子设备的通信效率。

图 2 示出根据本申请实施例的应用程序间通信方法的示例性应用场景。

如图 2 所示，本申请实施例的应用程序间通信方法可应用于电子设备，本申请的电子设备可以是智能手机、上网本、平板电脑、笔记本电脑、可穿戴电子设备(如智能手环、智能手表等)、TV、虚拟现实设备等等。

电子设备可包括虚拟机监控器和宿主机，其中宿主机可运行宿主机侧进程 1、宿主机侧进程 2，并可通过虚拟机监控器运行客户机 1、客户机 2。客户机 2 还可运行客户机 2 进程 1。

本申请实施例的应用程序间通信方法可以为第一虚拟装置和第二虚拟装置提供快速通信服务，以及为第一虚拟装置上的第一应用程序和第二虚拟装置上的第二应用程序之间提供快速通信服务，其中，第一虚拟装置可以是物理机（宿主机），第二虚拟装置可以是虚拟机（客户机 1 或客户机 2）。第一应用程序可以对应于宿主机侧进程 1 或宿主机侧进程 2，第二应用程序可以对应于客户机 2 进程 1。

其中，第一应用程序（或第二应用程序）在运行过程中，可能需要第二应用程序（或第一应用程序）运行对应的功能进行配合，此时第一应用程序（或第二应用程序）可发起向第二应用程序（或第一应用程序）通信请求。已发出的请求可通过第二虚拟装置（或第一虚拟装置）对应的命令队列（未示出）来承载。请求已发出的信息可由第一虚拟装置（或第二虚拟装置）发出核间中断来通知第二虚拟装置（或第一虚拟装置）。第二虚拟装置（或第一虚拟装置）根据核间中断，可以获取被命令队列承载的请求，并驱动其上的第二应用程序对该请求作出响应。从而实现第一应用程序和第二应用程序之间的快速通信。

图 2 中以第一虚拟装置是物理机，第二虚拟装置是虚拟机作为示例。本领域技术人员应理解，第一虚拟装置也可以是虚拟机，第二虚拟装置也可以是宿主机，本申请实施例对于第一虚拟装置和第二虚拟装置的具体类型不作限制。

图 3 示出根据本申请实施例的应用程序间通信方法的示例性流程示意图。

如图 3 所示，在一种可能的实现方式中，本申请提出一种应用程序间通信方法，所述方

法应用于电子设备，所述电子设备运行第一虚拟装置和第二虚拟装置，所述第一虚拟装置和所述第二虚拟装置中的一个是虚拟机，另一个是物理机，所述方法包括步骤S1-步骤S4：

步骤S1，所述第一虚拟装置根据第一应用程序发出的第一请求，向第一命令队列中写入第一命令，所述第一请求、所述第一命令包括所述第二虚拟装置上的第二应用程序的标识。

其中，第一应用程序可以是第一虚拟装置上的某一应用程序，第一请求可以是第一应用程序向第二虚拟装置上的第二应用程序通信的请求。第一命令队列可以是电子设备启动时即初始化好的、第二虚拟装置可访问的静态共享内存。第一命令可以储存在第一命令队列中，等待第二虚拟装置进行获取和处理。可选地，第一请求、第一命令还可以包括指示第二应用程序的运行方式的信息。

步骤S2，所述第一虚拟装置输出第一核间中断到所述第二虚拟装置所在的核心，以使所述第二虚拟装置能够获取所述第一核间中断。

核间中断是直接输出至需通信的对象所在的核心，因此允许不使用虚拟中断控制器实现物理机与虚拟机之间、不同虚拟机的应用程序之间的互相通知。第一虚拟装置输出第一核间中断到第二虚拟装置时，可认为是第一虚拟装置向第二虚拟装置的通信。在第二虚拟装置获取到第一核间中断时，可认为第一虚拟装置向第二虚拟装置的通信已完成。

步骤S3，所述第二虚拟装置根据所述第一核间中断从所述第一命令队列获取所述第一命令。

第二虚拟装置接收到核间中断时，可认为有新的命令写入第一命令队列，且该新写入的命令需第二虚拟装置进行处理。因此，第二虚拟装置可根据第一核间中断从第一命令队列处获取第一命令。

步骤S4，所述第二虚拟装置根据所述第一命令中、所述第二应用程序的标识运行所述第二应用程序。

第二虚拟装置可认为第一命令所包括的标识对应的应用程序即为需运行的应用程序。第一命令包括第二应用程序的标识，因此，第二虚拟装置根据第一命令，可以确定需运行的是第二应用程序。在第一命令包括指示第二应用程序的运行方式的信息时，第二虚拟装置可以按照第一命令的指示运行第二应用程序。此时可认为第二虚拟装置根据通信完成了对应响应。

根据本申请实施例的应用程序间通信方法，通过第一虚拟装置根据第一应用程序发出的第一请求向第一命令队列中写入第一命令，并输出第一核间中断到第二虚拟装置所在的核心，使得第二虚拟装置可以获取第一核间中断并根据第一核间中断从第一命令队列获取第一命令，可以实现第一虚拟装置到第二虚拟装置的通信；第一命令是根据来自第一应用程序的第一请求写入的，并包括第二虚拟装置上的第二应用程序的标识，使得第二虚拟装置根据第一命令中、第二应用程序的标识可以运行第二应用程序，从而能够实现第一虚拟装置的应用程序到第二虚拟装置的应用程序的通信。由于第一虚拟装置和所述第二虚拟装置中的一个是虚拟机，另一个是物理机，因此物理机和虚拟机之间使用将核间中断直接输出至需通信的对象所在的核心的方式完成通信，可以避免使用虚拟中断控制器，从而避免虚拟机的陷入陷出，可以实现虚拟机的业务和物理机的业务之间的快速通信，提高电子设备的通信效率。

图4示出根据本申请实施例的第一应用程序通知第二应用程序的示例性方法的示意图。

如图4所示，在一种可能的实现方式中，所述第一虚拟装置还包括第一代理模块，步骤S1包括：

所述第一代理模块接收所述第一应用程序发送的所述第一请求；

所述第一代理模块根据所述第一请求，向所述第一命令队列中写入所述第一命令；

步骤 S2 包括：

所述第一代理模块输出所述第一核间中断到所述第二虚拟装置所在的核心。

通过这种方式，可以使用第一代理模块完成与第一应用程序相关的部分工作，避免第一应用程序直接与第二虚拟装置本身进行数据交换，保证了第一应用程序运行的安全性，同时降低了第一应用程序的数据处理成本。

举例来说，第一虚拟装置上可包括至少一个应用程序，该至少一个应用程序可以对应于一个第一代理模块。第一代理模块可以为出于安全角度考虑或者为实现计算机资源管理而引入的代理实体，运行于较高权限级别，可以类比为 Linux 内核中的内核态。第一代理模块可以负责管理、监控第一虚拟装置上运行于较低权限级别的应用程序，以及解除第一虚拟装置上阻塞的应用程序（示例参见图 6）的阻塞状态等，至少一个应用程序可以是相互通信的实体。本申请实施例中，至少一个应用程序可将想要做的事情以请求的形式发送至第一代理模块，请求第一代理模块帮助操作。响应于请求，第一代理模块能够进行收发核间中断、写命令队列等操作，所以在第一代理模块管理第一应用程序的过程中，可以辅助进行第一应用程序的鉴权（示例参见下文图 10a、图 10b）、内存映射（示例参见下文图 7）等事项。

步骤 S1 中，第一应用程序发出的第一请求可以由第一代理模块接收。在第一请求为第一应用程序发起的、向第二虚拟装置上的第二应用程序通信的请求时，第一请求可以包括第二应用程序的标识。

第一虚拟装置根据第一应用程序发出的第一请求向第一命令队列中写入第一命令，可以是第一代理模块根据第一请求，向第一命令队列中写入第一命令。作为与第一请求对应的第一命令，在第一请求包括第二应用程序的标识时，第一命令也可以包括第二应用程序的标识。

当第一虚拟装置希望发送第一请求给第二虚拟装置时，可使用第一命令队列，按照预设的格式（如[命令类型，命令参数]）将对应于第一请求的第一命令写到第一命令队列中。除可被第一代理模块访问之外，第一命令队列可以设置为可被第二虚拟装置上的应用程序的代理模块（例如后文所述的第二代理模块）访问。

第一核间中断也可以由第一代理模块产生。步骤 S2 中，第一虚拟装置输出第一核间中断到第二虚拟装置所在的核心，可以是第一代理模块输出第一核间中断到第二虚拟装置所在的核心，从而实现第一虚拟装置通知第二虚拟装置。

如果第一应用程序的运行需第二应用程序配合时，第一应用程序不必等待第二应用程序运行完毕后才能继续运行，即不必阻塞时，则第一应用程序发出第一请求之后，即可继续运行，而不进入阻塞状态。在此情况下，第二应用程序运行结束后，可不通知第一应用程序。

下面介绍本申请实施例通过第一核间中断的发送和接收实现第一虚拟装置和第二虚拟装置之间的通知的原理。

出于高性能的需求，宿主机和客户机之间进行通信时，都需要直接发送核间中断给对方，且此流程中不需要客户机陷入陷出。然而在 arm 平台上，如果使能虚拟中断控制器（vgic）实现中断的发送，无论是客户机想要发送中断通知宿主机的时候，还是宿主机发送中断通知客户机的时候，理论上都需要客户机陷出。

基于此，本申请实施例的应用程序间通信方法可实现不使能虚拟中断控制器的情况下的

核间中断的收发。由于没有使能虚拟中断控制器，因此，无论是第一虚拟装置侧发出的核间中断还是第二虚拟装置侧发出的核间中断，都可以直接发送给接收核间中断时正在运行的实体。以下以第一虚拟装置作为实体接收核间中断为例进行描述。

图 5a 示出根据本申请实施例的第一虚拟装置作为实体接收核间中断的一个示例。

如图 5a 所示，假设第一虚拟装置作为物理机，且物理机还用于运行至少一个虚拟机以及虚拟机对应的虚拟机管理线程（客户机进程），物理机在电子设备的某一核心上。物理机正在运行时接收到核间中断，如果核间中断是电子设备的另一核心发送给物理机的，则核间中断可由物理机接收并处理。如果核间中断是发送给物理机运行的某一虚拟机的，则物理机可以通过中断处理器将此核间中断注入给对应的虚拟机，由虚拟机进行处理。注入的方式可以通过现有技术来实现，例如在通过硬件方式注入时，可以写虚拟中断控制器上的寄存器（未示出），又例如通过软件方式注入时，可以更改虚拟机的寄存器上下文。本申请对于物理机将核间中断注入给对应的虚拟机的具体方式不作限制。

图 5b 示出根据本申请实施例的第一虚拟装置作为实体接收核间中断的另一个示例。

如图 5b 所示，假设第一虚拟装置作为虚拟机，且虚拟机由一物理机运行，该物理机还运行虚拟机对应的虚拟机管理线程（客户机进程），该物理机设置在电子设备的某一核心上。虚拟机正在运行时接收到核间中断，如果核间中断是电子设备的另一核心发送给虚拟机的，则核间中断可由虚拟机接收，直接处理核间中断即可；如果核间中断是发送给运行虚拟机的物理机的，则虚拟机可通过中断处理器发出超级调用（hypercall）指令将中断发送至物理机。物理机根据超级调用指令，可以主动进行中断处理。

结合图 5a 和图 5b 的相关描述可知，对于正运行的虚拟机，在收到向自己发出的核间中断时不需要陷出的动作。同时，因为没有使能虚拟中断控制器，所以虚拟机也可以直接发送核间中断给物理机所在核心。通过这种方式，使得核间中断不必使能虚拟中断控制器也可以在物理机和虚拟机之间发送，且虚拟机也不必陷入陷出，从而实现虚拟机与物理机之间的中断的直接通知。

如图 4 所示，在一种可能的实现方式中，所述第二虚拟装置还包括第二代理模块，步骤 S3 包括：

所述第二代理模块根据所述第一核间中断从所述第一命令队列获取所述第一命令；

步骤 S4 包括：

所述第二代理模块根据所述第一命令中、所述第二应用程序的标识运行所述第二应用程序。

通过这种方式，可以使用第二代理模块完成与第二应用程序相关的部分工作，避免第二应用程序直接与第一虚拟装置本身进行数据交换，保证了第二应用程序运行的安全性，同时降低了第二应用程序的数据处理成本。

举例来说，第二虚拟装置上可包括至少一个应用程序，该至少一个应用程序可以对应于一个第二代理模块。类似于第一代理模块，第二代理模块也可以为出于安全角度考虑或者为实现计算机资源管理而引入的代理实体，运行于较高权限级别，可以类比为 Linux 内核中的内核态。第二代理模块可以负责管理、监控第二虚拟装置上运行于较低权限级别的应用程序，以及解除第二虚拟装置上阻塞的应用程序的阻塞状态等，至少一个应用程序可以是相互通信的实体。本申请实施例中，至少一个应用程序可将想要做的事情以请求的形式发送至第二代

理模块，请求第二代理模块帮助操作。响应于请求，第二代理模块能够进行收发核间中断、写命令队列等操作，所以在第二代理模块管理第二应用程序的过程中，可以辅助进行第二应用程序的鉴权（示例参见下文图 10a、图 10b）、内存映射（示例参见下文图 7）等事项。在引入第一代理模块和第二代理模块的基础上，第一虚拟装置侧的应用程序和第二虚拟装置侧的应用程序即可做到相互通信。

步骤 S3 中，第二代理模块可以通过访问第一命令队列获取第一命令。即第一命令队列可以使设置为仅第二代理模块和第一代理模块才有资格访问，对于其他实体不可见。

参见上文描述，第一命令可包括第二应用程序的标识，在步骤 S4 中，第二代理模块根据该标识，可以确定该标识对应的应用程序也即第二应用程序是需运行的应用程序，并运行第二应用程序。相当于第二应用程序接收到了来自第一应用程序的通知，并做出了响应。且该通知过程不需虚拟机的陷入陷出，从而实现了虚拟机与物理机之间的应用程序的快速通知。

图 6 示出根据本申请实施例的第一应用程序通知第二应用程序之后接收到第二应用程序的通知的示例性方法的示意图。

如图 6 所示，在一种可能的实现方式中，所述第一请求和所述第一命令还包括所述第一应用程序的阻塞信息，

所述第一代理模块接收所述第一应用程序发送的所述第一请求之后，所述方法还包括：

所述第一应用程序进入阻塞状态；

在接收到第二核间中断时，所述第一代理模块根据所述第二核间中断从第二命令队列获取第二命令，所述第二核间中断在所述第二应用程序运行结束后产生；

在所述第二命令包括所述第一应用程序的标识时，所述第一代理模块解除所述第一应用程序的阻塞状态。

通过这种方式，对于发出向第二应用程序通信的请求后阻塞的第一应用程序，可以实现第二应用程序到第一应用程序的快速通知，阻塞的第一应用程序可以通过该通知解除阻塞状态，使得可以保证第一应用程序可以快速地继续运行。

举例来说，如果第一应用程序的运行需第二应用程序配合，且第一应用程序必须等待第二应用程序运行完毕后才能继续运行，则第一应用程序可发出包括第一应用程序的阻塞信息的第一请求。阻塞信息可以是指示应用程序进入阻塞状态的指令。第一应用程序的阻塞信息，可以是指示第一应用程序进入阻塞状态的指令。在发出第一请求之后，第一应用程序可以在对应的信号量（未示出）上阻塞，即进入阻塞状态。在阻塞状态下，在接收到来自第二应用程序的通知之后，可以继续运行。

在步骤 S1 中，第一代理模块接收第一应用程序发送的第一请求之后，在向第一命令队列中写入第一命令时，第一命令也可以包括第一应用程序的阻塞信息，即第一请求、第一命令可同时包括第一应用程序的标识和第一应用程序的阻塞信息。结合上文描述可知，第一命令可由第二代理模块获取，并根据第一命令中的标识运行第二应用程序，在第二应用程序运行结束之后，根据第一应用程序的阻塞信息，第二应用程序可以反向通知第一应用程序，其通知方式可以与第一应用程序通知第二应用程序的方式相同。例如第二应用程序发送第二请求，第二请求可包括第一应用程序的标识；第二代理模块根据第二请求将第二命令写入第二命令队列，并发送第二核间中断到第一虚拟装置所在的核心，第二命令可包括第一应用程序的标识。其具体的通知方式可以参见下文中图 6 的相关描述。对于第一虚拟装置侧，则可体现为，

第一虚拟装置接收到第二核间中断时，第一代理模块根据第二核间中断从第二命令队列获取第二命令；根据第二命令包括的第一应用程序的标识，第一代理模块解除第一应用程序的阻塞状态，此时第一应用程序可继续运行。第二核间中断的接收方式可以参见上文及图 5a 和图 5b 的相关描述，在此不再赘述。

进一步地，第一代理模块在代理多个应用程序时，如果第一应用程序必须等待第二应用程序运行完毕后才能继续运行，则第一请求、第一命令还可包括第一应用程序的标识，即第一请求、第一命令可同时包括第一应用程序的标识、第二应用程序的标识、第一应用程序的阻塞信息。以使第二代理模块根据第一命令中第一应用程序的标识确定第二应用程序运行结束之后，可进一步确定第二应用程序应通知的是第一代理模块代理的多个应用程序中的第一应用程序，以保证第一应用程序可以快速地恢复正常运行的状态。

如图 6 所示，在一种可能的实现方式中，所述第二虚拟装置根据所述第一命令中、所述第二应用程序的标识运行所述第二应用程序之后，所述方法还包括：

所述第二应用程序运行结束后，所述第二代理模块接收所述第二应用程序发送的第二请求，所述第二请求包括所述第一应用程序的标识；

所述第二代理模块根据所述第二请求，向所述第二命令队列中写入所述第二命令，所述第二命令包括所述第一应用程序的标识；

所述第二代理模块输出所述第二核间中断到所述第一虚拟装置所在的核心，以使所述第一虚拟装置能够获取所述第二核间中断。

通过这种方式，使得第二代理模块可以确定第二代理模块运行结束后需通知的是第一应用程序，使得第一代理模块根据第二命令可以解除的是第一应用程序的阻塞状态，以保证第一应用程序解除阻塞状态时，第二应用程序确实已经运行结束，从而能满足第一应用程序等待第二应用程序运行完毕后继续运行的需求。

举例来说，如果第一命令包括第一应用程序的阻塞信息，则根据第一命令，第二代理模块可以确定第一应用程序已经处于阻塞状态，且阻塞状态需等待第二应用程序运行结束后的通知来解除。在此情况下，第二代理模块调度第二应用程序运行结束后，第二应用程序可发出第二请求，由于第二应用程序是根据第一应用程序的通知运行的，因此第二请求可包括第一应用程序的标识。

根据第二请求，第二代理模块可以向第二命令队列中写入第二命令。作为与第二请求对应的第二命令，在第二请求包括第一应用程序的标识时，第二命令也可以包括第一应用程序的标识。可选地，第二请求和第二命令还可以包括指示第一应用程序的运行方式的信息。

第二命令队列可以是第一虚拟装置和第二虚拟装置之间的、一块第一虚拟装置启动时即初始化好的静态共享内存。当第二虚拟装置希望发送第二请求给第一虚拟装置时，可使用第二命令队列，按照预设的格式（如[命令类型，命令参数]）将对应于第二请求的第二命令写到第二命令队列中。第二命令队列可以设置为可被第一虚拟装置上的一些模块（例如上文所述的第一代理模块）访问。

在写入第二命令到第二命令队列之后，第二代理模块可以输出第二核间中断到第一虚拟装置所在的核心，从而实现第二虚拟装置通知第一虚拟装置。第二核间中断的输出方式可以参见上文及图 5a 和图 5b 的相关描述，在此不再赘述。

进一步地，第二代理模块代理多个应用程序在代理多个应用程序时，如果第一应用程序

必须等待第二应用程序运行完毕后才能继续运行，则第二请求、第二命令还可包括第二应用程序的标识，即第二请求、第二命令可同时包括第一应用程序的标识和第二应用程序的标识。以使第二代理模块可以根据第二请求中第二应用程序的标识可以确定其代理的多个应用程序中的第二应用程序运行结束，从而能执行根据第二请求向第二命令队列中写入第二命令的步骤。同时第一代理模块可以根据第二命令中第二应用程序和第一应用程序的标识，确定由第一应用程序发起的、第一应用程序向第二虚拟装置上的第二应用程序通信的请求已经处理完毕，第二应用程序已经响应于该请求运行结束，再解除第一应用程序的阻塞状态。以保证第一应用程序解除阻塞状态恢复正常运行的状态时，第二应用程序确实已经运行结束。

以上所述的第一应用程序的标识可以由第一应用程序向其代理也即第一代理模块申请，用于身份识别和阻塞状态的解除。第一代理模块可以将第一应用程序的标识与任一信号量（未示出）关联起来，使得第一应用程序可以在对应的信号量上阻塞。类似地，第二应用程序的标识可以由第二应用程序向其代理也即第二代理模块申请，用于身份识别和阻塞状态的解除。第二代理模块可以将第二应用程序的标识与任一信号量（未示出，不同于第一应用程序对应的信号量）关联起来，使得第二应用程序可以在对应的信号量上阻塞。

以上仅示出第一应用程序阻塞的示例。本领域技术人员应理解，第二应用程序的运行需除第二虚拟装置之外的其他物理机的对应的应用程序配合时，如果第二应用程序必须等待对应的应用程序运行完毕后才能继续运行，则第二应用程序也可在第二代理模块为其关联的信号量上阻塞，阻塞的方式以及阻塞状态解除的方式可以参见第一应用程序阻塞的方式以及阻塞状态解除的方式的相关描述，在此不再赘述。

在一种可能的实现方式中，所述第一代理模块根据所述第一请求，向所述第一命令队列中写入所述第一命令，包括：

所述第一代理模块根据所述第一请求，判断所述第二应用程序是否具有通知权限；

在所述第二应用程序具有通知权限时，向所述第一命令队列中写入所述第一命令。

通过这种方式，可以保证接收到通知的应用程序是具有通知权限的应用程序，保证通知的安全性。

举例来说，在一些应用场景下，例如微内核场景，对于互相通信的应用程序的权限可能有一定需求，需要对应用程序进行鉴权，以保证通信的安全性。因此，在向第一命令队列中写入第一命令之前，第一代理模块可以对第二应用程序的权限进行判断。第一请求可以包括第二应用程序的标识，因此，第一代理模块根据第一请求，可以确定第二应用程序是第一应用程序想要请求运行（通知）的应用程序。在此情况下，第一代理模块可以先确定第二应用程序是否具有通知权限。在第二应用程序具有通知权限时，可认为第二应用程序是可以通信的对象，再向第一命令队列中写入第一命令。如果第二应用程序不具有通知权限时，可认为第二应用程序不是可以通信的对象，可不向第一命令队列中写入第一命令。

通知权限可以直接由第一虚拟装置上的第一代理模块确定。例如，对于第二代理模块代理的各应用程序，在准备使用本申请实施例的应用程序间通信方法进行通信时，可以向第二代理模块申请对应的标识，用于身份识别和阻塞状态的解除时使用。在某一应用程序申请标识的时候可以对该应用程序进行一次授权。以第二代理模块代理的第二应用程序为例，在第二应用程序向第二代理模块申请标识之后，可例如由第二代理模块使用密码学手段（如签名）或者静态表配置的方式进行应用程序合法性的检查。如果第二应用程序是不合法的应用程序，

第二代理模块可不为第二应用程序提供标识，所以后续第二应用程序就无法使用本申请实施例的应用程序间通信方法；或者，第二代理模块也可为第二应用程序提供标识，但记录第二应用程序不具备通知权限。反之，如果第二应用程序是合法的应用程序，第二代理模块可为第二应用程序提供标识，以使第二应用程序可以使用本申请实施例的应用程序间通信方法；第二代理模块可以记录第二应用程序具备通知权限。第二代理模块可以将第二应用程序具备通知权限的信息通过第二虚拟装置与第一虚拟装置的接口（例如下文所述的第一进程模块）传递给第一虚拟装置侧的第一代理模块。

第一代理模块第一次判断第二应用程序是否具有通知权限之后，在后续需要再次判断第二应用程序是否具有通知权限时，可以直接使用第一次判断的判断结果，可以不必重复进行判断。

一些情况下，第一应用程序和第二应用程序之间除了互相通知，还需要传递很多数据消息，如果仍然使用第一命令队列和第二命令队列进行数据的传递，对于可传递的数据的种类、数量等等都会带来限制。基于此，本申请实施例还提出利用应用程序间通知的机制为第一应用程序和第二应用程序之间动态地创建共享内存，使其可以作为第一应用程序和第二应用程序之间的专用数据通道，可以用于传递种类、数据量都比较多的数据。下面介绍本申请实施例的专用数据通道的示例性建立方法。

图 7 示出根据本申请实施例建立专用数据通道的示例性示意图。

如图 7 所示，在一种可能的实现方式中，所述第一虚拟装置根据所述第一应用程序发出的第一请求，向第一命令队列中写入第一命令之前，所述方法还包括：

响应于第一应用程序的申请，所述第一虚拟装置创建共享内存，所述第一代理模块建立所述第一应用程序的地址与所述共享内存的地址的关联关系；所述第一请求和所述第一命令还包括所述共享内存的参数；

在所述第二命令包括所述第一应用程序的标识时，所述第一代理模块解除所述第一应用程序的阻塞状态，包括：

在所述第二命令包括所述第一应用程序的标识以及通道建立成功的信息时，所述第一代理模块解除所述第一应用程序的阻塞状态。

通过这种方式，使得可以通过共享内存作为第一应用程序和第二应用程序之间的专用数据通道，相比仅采用命令队列的形式进行数据传递的方案，可以传递更丰富的数据。同时共享内存可传递的数据量比较大，可以减少数据的拷贝次数，提升数据的传输效率。

举例来说，第一应用程序可以向第一虚拟装置的核心申请创建一块共享内存，该申请得到第一虚拟装置的核心的引用后，第一虚拟装置的核心可以分配一块物理内存页面作为共享内存（即第一虚拟装置创建共享内存），物理内存页面的真实地址与共享内存的虚拟地址相关联。第一代理模块可以建立第一应用程序的地址与共享内存的地址的关联关系（<1>），因此，第一应用程序就可以使用该共享内存。由于共享内存关联于物理内存页面，因此，第一应用程序使用该共享内存存取数据时，可满足多种类的、数据量较大的数据的存取。图 7 中，HVA 表示对应于物理内存页面的虚拟的共享内存地址。

在第一应用程序具备可使用的共享内存的情况下，第一请求和第一命令还可包括共享内存的参数，例如共享内存的名称、大小等等。此时第一请求可以是第一应用程序向第二虚拟装置上的第二应用程序通信、以及创建第一应用程序和第二应用程序之间的专用数据通道的

请求。

第二虚拟装置上，第二代理模块根据第一命令，在驱动第一应用程序运行的同时，还可以为第二应用程序和第一应用程序建立专用数据传输通道（<4><5>），其示例性方式可以参见下文中图 7 的相关描述。第二请求可以在专用数据通道建立成功之后由第二应用程序发出。第二请求可以包括通道建立成功的信息。因此，第二命令也可以包括通道建立成功的信息，即第二命令可以包括第一应用程序的标识以及通道建立成功的信息。

在第二命令包括第一应用程序的标识时，第一代理模块解除第一应用程序的阻塞状态，可以是在第二命令包括第一应用程序的标识以及通道建立成功的信息时，第一代理模块解除第一应用程序的阻塞状态。此时第一应用程序和第二应用程序就可以通过专用数据通道互相传递数据信息。可选地，在通道建立成功时，通道建立成功的信息也可以通过建立好的专用数据通道进行传递，本公开对于通道建立成功的信息的具体传递方式不作限制。

如图 7 所示，在一种可能的实现方式中，所述第二代理模块根据所述第一命令中、所述第二应用程序的标识运行所述第二应用程序，包括：

所述第二代理模块根据所述第一命令，判断所述第二应用程序是否具有访问所述共享内存的权限；

所述第二代理模块运行所述第二应用程序，并在所述第二应用程序具有访问所述共享内存的权限时，建立所述第二应用程序的地址与所述共享内存的地址的关联关系；所述第二命令还包括通道建立成功或通道建立失败的信息。

通过这种方式，使得专用数据通道建立前，第二代理模块可以确定第二应用程序是否具备访问共享内存的权限，专用数据通道在第二应用程序具备访问共享内存的权限时建立，以保证专用数据通道的建立的安全性。通过第二命令将通道建立成功或通道建立失败的信息传递给第一虚拟装置侧，以使第一虚拟装置的第一应用程序可以确定其请求的执行情况。

举例来说，在第一命令包括第二应用程序的标识以及共享内存的参数时，第二代理模块根据第一命令，可以确定第一应用程序发起了向第二虚拟装置上的第二应用程序通信、以及创建第一应用程序和第二应用程序之间的专用数据通道的请求。在一些场景下，例如微内核场景，对于应用程序的权限可能有一定需求，需要对应用程序进行鉴权，以保证数据传输的安全性。在此情况下，第二代理模块可以先运行第二应用程序，同时确定第二应用程序是否具有访问所述共享内存的权限。在第二应用程序具有访问所述共享内存的权限时，可认为第二应用程序是可以作为专用数据通道的使用者的，此时可以分配虚拟共享内存地址（GPA），并利用超级调用指令请求虚拟机监控器或者物理机内核建立 2 阶段页表，即建立虚拟共享内存地址 GPA 与共享内存（物理）地址的关联关系（<4>）。之后第二代理模块再将第二应用程序的地址（GVA）映射到虚拟共享内存地址 GPA 上（<5>），完成建立第二应用程序的地址与共享内存的地址的关联关系。共享内存的位置、大小的同步可以借助第一命令队列和第二命令队列实现。由上文描述可知，共享内存的地址与第一应用程序的地址也具有关联关系，因此，相当于第一应用程序的地址与第二应用程序的地址相关联。即第一应用程序和第二应用程序之间的专用数据通道建立成功，第一应用程序和第二应用程序可以使用共享内存完成相互的数据传输。第二命令还可包括通道建立成功的信息，以通知第一应用程序其请求的专用数据通道已经成功建立。

如果第二应用程序不具有访问所述共享内存的权限时，可认为第二应用程序不可以作为

专用数据通道的使用者，可不建立第二应用程序的地址与共享内存的地址的关联关系。第二命令还可包括通道建立失败的信息，以通知第一应用程序其请求的专用数据通道未能成功建立。

可选地，如果物理机侧有接口支持，也可以基于上述图 7 的流程动态扩缩共享内存的能力。例如在不需传输数据或者需要传输的数据量远小于已经创建的共享内存的大小时，已经建立的专用数据通道还可以动态地删除或者进行大小的动态调整，以保证资源部署的灵活性，提升内存使用效率。

由于本申请实施例的应用程序间通信方法依赖于虚拟机和物理机之间的核间中断收发实现互相通知，因此，如果某两个应用程序之间多次通知，产生的命令的个数可能较多，此时可以通过以下的控制通道对核间中断进行汇聚，以降低核间中断的收发次数。

在一种可能的实现方式中，所述共享内存还作为控制通道，所述控制通道包括第一数据位和第二数据位，所述第一数据位和所述第二数据位的状态由所述第一代理模块和所述第二代理模块控制；

步骤 S2 之前，所述方法还包括：

所述第一代理模块控制所述第一数据位处于置位状态、第二数据位处于非置位状态；

所述第一代理模块输出所述第一核间中断到所述第二虚拟装置所在的核心，包括：

所述第一代理模块确定所述第一应用程序和所述第二应用程序之间的控制通道的第一数据位处于置位状态、第二数据位处于非置位状态时，所述第一代理模块输出第一核间中断到所述第二虚拟装置所在的核心。

通过控制通道专用于控制核间中断的发送及接收，可以降低核间中断的收发次数，降低应用程序间通信方法的实现复杂度。

举例来说，可选地，可以设置第一应用程序和第二应用程序之间的某一专用数据通道作为控制通道，此时控制通道也成为第一应用程序和第二应用程序之间的私有控制通道。图 8 示出根据本申请实施例的控制通道的一个示例。

如图 8 所示，第一应用程序、第三应用程序可以是第一虚拟装置侧的应用程序，第二应用程序、第四应用程序可以是第二虚拟装置侧的应用程序，在图 8 的示例中，在电子设备上可以包括第一应用程序和第二应用程序之间的控制通道，第一应用程序和第四应用程序之间的控制通道、以及第三应用程序和第二应用程序之间的控制通道。

以第一应用程序和第二应用程序之间的控制通道为例，该控制通道可以至少包括第一数据位和第二数据位，二者的状态可由第一代理模块和第二代理模块控制。其中，第一数据位和第二数据位的状态可包括置位状态和非置位状态。该通道的第一数据位的置位状态可以表示第一应用程序有发送给第二应用程序的请求，在此情况下，该通道的第二数据位的置位状态可以表示第二应用程序正在处理该请求；该通道的第二数据位的非置位状态可以表示第二应用程序尚未处理该请求；该通道的第一数据位的非置位状态可以表示第一应用程序没有发送给第二应用程序的请求。

由于控制通道本质上是第一应用程序和第二应用程序之间共享内存，因此第一数据位和第二数据位的状态对于第一代理模块和第二代理模块可以是同步的，例如第一代理模块控制第一数据位为置位状态后，第二代理模块如获取第一数据位的状态，则获取到的可以是置位状态；此时第二代理模块控制第一数据位为非置位状态后，第一代理模块如获取第一数据位

的状态，则获取到的可以是非置位状态。

因此，在第一虚拟装置需发送第一核间中断至第二虚拟装置时，在步骤 S2 执行之前，第一代理模块可以先控制该第一应用程序和第二应用程序之间的控制通道的第一数据位处于置位状态、第二数据位处于非置位状态，表示第一应用程序有发送给第二应用程序的请求，且第二应用程序尚未处理该请求。在此情况下，第一代理模块再输出第一核间中断到第二虚拟装置。

在一种可能的实现方式中，所述第二代理模块根据所述第一核间中断从所述第一命令队列获取所述第一命令，包括：

所述第二代理模块根据所述第一核间中断，遍历所述第二虚拟装置上、所述第二代理模块对应的全部应用程序关联的控制通道；

在确定所述第一应用程序和所述第二应用程序之间的控制通道的第一数据位处于置位状态时，所述第二代理模块从第一命令队列获取所述第一命令。

通过第二代理模块对于其对应的全部应用程序关联的控制通道的第一数据位均进行检查，可以确定是哪一通道对应的应用程序有请求发出；通过第二代理模块根据第一数据位的状态确定是否需获取命令，可以避免命令的重复获取，可以提高第二代理模块的工作效率。

举例来说，第二虚拟装置上，第二代理模块可代理至少一个应用程序，每一应用程序和其他虚拟机或者物理机上的应用程序之间都可能关联一个控制通道，即第二代理模块可对应于多个控制通道。在核间中断到来时，第二代理模块可能仅可以确定有请求需其代理的某一应用程序处理，而对于该核间中断是根据哪一应用程序的请求发出、具体需要哪一应用程序处理可能是未知的。

在此情况下，第二代理模块根据第一核间中断从第一命令队列获取所述第一命令，可以是第二代理模块先根据第一核间中断，遍历第二虚拟装置上、第二代理模块对应的全部应用程序关联的控制通道，确定各控制通道的第一数据位的状态。在确定第一应用程序和第二应用程序之间的控制通道的第一数据位处于置位状态时，可认为第一应用程序有发送给第二应用程序的请求，再从第一命令队列获取第一命令。如果确定第一应用程序和第二应用程序之间的控制通道的第一数据位处于非置位状态时，可认为第一应用程序没有请求需第二应用程序处理，不必从第一命令队列获取第一命令。

在一种可能的实现方式中，步骤 S4 之后，所述方法还包括：

所述第二代理模块控制所述第一应用程序和所述第二应用程序之间的控制通道的第一数据位处于非置位状态、第二数据位处于置位状态；

针对所述第一应用程序和所述第二应用程序之间的控制通道，所述第二应用程序运行结束后，所述第二代理模块确定该控制通道的第一数据位处于非置位状态时，控制该控制通道的第二数据位处于非置位状态；

所述第二代理模块确定该控制通道的第一数据位处于置位状态时，重新从所述第一命令队列获取命令，并根据新获取的命令中的第二应用程序的标识调度所述第二应用程序运行。

通过这种方式，可以在第一应用程序和所述第二应用程序之间的控制通道的请求处理完毕后，继续处理第一应用程序后续发起的新请求，而不必先获取核间中断后再对新请求进行处理，可以降低核间中断的收发次数，节约数据传输成本。

举例来说，在第二代理模块从第一命令队列获取第一命令之后，可以执行步骤 S4 运行第

二应用程序，对第一应用程序发送给第二应用程序的请求进行处理。此时可认为第二应用程序已经处理该请求。第二代理模块可以控制第一应用程序和所述第二应用程序之间的控制通道的第一数据位处于非置位状态、第二数据位处于置位状态。在第二应用程序运行的过程中，第一应用程序还可能继续发起请求，这一动作可能导致该控制通道的第一数据位在第一代理模块的控制下处于置位状态。因此，在第二应用程序运行结束之后，第二代理模块可以检查第一应用程序和第二应用程序之间的控制通道的第一数据位，如确定该控制通道的第一数据位仍处于非置位状态，可认为第一应用程序没有进一步向第二应用程序发起请求，可以清除第二数据位的置位状态，即控制该控制通道的第二数据位处于非置位状态。如确定该控制通道的第一数据位处于置位状态，可认为第一应用程序进一步向第二应用程序发起了其他请求，可以重新从第一命令队列获取命令，并根据新获取的命令中的应用程序的标识调度对应的应用程序运行。

在按照某一控制通道的请求完成调度应用程序运行之后，确定该控制通道的第一数据位处于非置位状态，第二代理模块可以继续其他控制通道的第一数据位、第二数据位的状态查看和请求处理。其具体实现方式可以参见上文中与第一应用程序和所述第二应用程序之间的控制通道相关的描述，在此不再赘述。

参见上文所述，控制通道和专用数据通道都是第一应用程序和所述第二应用程序之间通道，且二者的职能不同，所以可以将第一应用程序和所述第二应用程序之间的控制通道和专用数据通道绑定起来，控制通道用来进行核间中断的汇聚，专用数据通道用来承载通信数据。专用数据通道又可以和数据发送方（如上文所述的第一应用程序）的标识和信号量绑定起来，用来让数据发送方阻塞和解除阻塞。图 9 示出根据本申请实施例的第一应用程序和所述第二应用程序之间的控制通道、专用数据通道、标识和信号量进行绑定的示例。综上所述，结合标识/信号量、控制通道、专用数据通道和第一命令队列、第二命令队列，本申请实施例的应用程序间通信方法可以成为可中断汇聚的高性能快速通知、数据传输方案。

上文中提及了在一些场景下，例如微内核场景，对于通信的安全性可能有需求，因此可进一步确定第二应用程序的通知权限以及访问所述共享内存的权限。其中，确定第二应用程序是否具备通知权限的相关描述已经介绍过，下面介绍确定第二应用程序是否具备访问所述共享内存的权限的示例性方法。图 10a 和图 10b 分别示出根据本申请实施例确定第二应用程序是否具备访问所述共享内存的权限的示例性方法示意图。

如图 10a 所示，在一种可能的实现方式中，在所述第一虚拟装置是物理机、所述第二虚拟装置是虚拟机时，所述第一虚拟装置还包括鉴权模块和第一进程模块，所述第二代理模块根据所述第一命令，判断所述第二应用程序是否具有访问所述共享内存的权限，包括：

所述第二代理模块将所述第一命令中、所述第二应用程序的标识转发至所述第一虚拟装置上的第一进程模块；

所述第一进程模块确定与所述第二应用程序的标识对应的鉴权标识，并向所述鉴权模块发送权限信息获取请求，所述权限信息获取请求包括所述鉴权标识；

所述第一进程模块接收来自所述鉴权模块的所述第二应用程序的权限信息，并传输至所述第二代理模块；

所述第二代理模块根据所述第二应用程序的权限信息，判断所述第二应用程序是否具有访问所述共享内存的权限。

在第一虚拟装置是物理机时，第一进程模块可作为虚拟机侧和物理机侧的接口，使得虚拟机侧的应用程序的相关信息可以通过第一进程模块传输至物理机侧，从而可以利用物理机侧的鉴权模块为虚拟机侧的应用程序进行鉴权。通过鉴权模块进行鉴权，可以进一步提高数据传输的安全性。

举例来说，现有技术中，物理机的微内核拥有极强的安全性，可以使用诸如形式化验证的手段用数学手段证明其可靠性，微内核拥有针对内核对象的鉴权模块，鉴权模块可例如支持现有技术的能力安全机制（capability 机制）。在该能力安全机制中，鉴权标识可以当做一种特殊的对象，每一个应用程序都需要拥有对方的鉴权标识才能访问特定的内核对象，或者执行特定的流程。因此，对于本申请实施例的虚拟机上的应用程序，也可以借助鉴权模块进行鉴权。

如图 10b 所示，假设第一虚拟装置是物理机（宿主机），则鉴权模块可以是设置在物理机侧的模块。由于设置在同一物理机上，因此对于鉴权模块而言，物理机侧的各应用程序是位于可信边界之内的，物理机侧的各应用程序可以直接通过鉴权模块的能力安全机制进行鉴权。假设第二虚拟装置是虚拟机，则对于设置在虚拟机上的应用程序，想要通过鉴权模块鉴权时，需要借助于物理机上的第一进程模块。

第一进程模块可以负责处理物理机侧、对应于虚拟机的进程，该对应于虚拟机的进程本质上可以是一个普通的用户态线程，所以其本身同样需要通过鉴权模块进行权限管理。只有第一进程模块先得到了访问某些资源（例如共享内存）的权限后，才能给虚拟机内部的众多应用程序授权。

而针对虚拟机内部的应用程序，由于鉴权模块在物理机侧，物理机侧是无法感知到虚拟机内部的应用程序的区别的，所以可以由第一进程模块协助物理机为虚拟机内部的应用程序进行功能的维护，可以选择将每个虚拟机内部的应用程序都作为虚拟机内核的一个“实例”，即可以获得能力安全机制的实体。以第二应用程序为例，当第二应用程序需要鉴权时，需要经过第二代理模块，第二代理模块会将第二应用程序的标识传递至第一进程模块，由第一进程模块确定与第二应用程序的标识对应的鉴权标识（得到鉴权标识<->第二应用程序的表示的映射），并向鉴权模块发送权限信息包括鉴权标识的权限信息获取请求。

在此情况下，鉴权模块可以识别鉴权标识，并确定鉴权标识对应的应用程序的权限信息，该鉴权标识对应的应用程序即第二应用程序。鉴权模块可以发送确定的第二应用程序的权限信息到第一进程模块。第一进程模块第二应用程序的权限信息后，可以传输至第二代理模块，第二代理模块根据第二应用程序的权限信息，判断第二应用程序是否具有访问所述共享内存的权限。第二代理模块第一次判断第二应用程序是否具有访问所述共享内存的权限之后，在后续需要再次判断第二应用程序是否具有访问所述共享内存的权限时，可以直接使用第一次判断的判断结果，可以不必重复进行判断。

可选地，第二应用程序的通知权限也可以由鉴权模块鉴权，例如第二代理模块为第二应用程序提供标识时，可以将第二应用程序的身份信息和其标识经由第一进程模块告知鉴权模块，让鉴权模块了解到第二应用程序是合法的实体，以确定二应用程序具备通知权限。鉴权模块鉴权第二应用程序的通知权限的示例性实现方式可以与鉴权模块鉴权第二应用程序的访问所述共享内存的权限的方式相同，在此不再赘述。

可选地，本申请实施例的应用程序间通信方法也可以用于两个虚拟机之间的快速通信，

通过物理机的内核为需通信的两个应用程序分别所属的两个虚拟机都进行鉴权，同时使共享内存于两台虚拟机之间可见即可。此时，可由物理机为该两个虚拟机进行专用数据通道的初始化、申请等工作。

图 11 示出根据本申请实施例的应用程序间通信方法的一个具体示例。该示例可实现位于宿主机侧的 DB 客户端和客户机侧的 DB 服务器的快速通信。在图 11 的示例中，宿主机对应于第一虚拟装置，客户机对应于第二虚拟装置。宿主机侧，代理模块 A 对应于第一代理模块，DB 客户端对应于第一应用程序。客户机侧，代理模块 B 对应于第二代理模块，DB 服务器对应于第二应用程序。

如图 11 所示，首先，在应用程序间通信方法执行之前，在宿主机侧，DB 客户端已经向代理模块 A 申请了自己的标识，代理模块 A 也已经将 DB 客户端的信息传输给宿主机侧的鉴权模块。在客户机侧，DB 服务器已经向代理模块 B 申请了自己的标识，代理模块 B 也已经将 DB 服务器的信息经宿主机侧的第一进程模块传输给鉴权模块。且 DB 客户端已获得 DB 服务器的标识。DB 客户端已经得到一块共享内存的使用权限。

在应用程序间通信方法执行时，在宿主机侧，DB 客户端向代理模块 A 发出第一请求，第一请求可以是 DB 客户端和 DB 服务器之间的专用数据通道建立请求。第一请求可以包括 DB 客户端的标识、DB 服务器的标识、DB 客户端的阻塞信息、共享内存的参数。发出第一请求后，DB 客户端进入阻塞状态。

根据第一请求，代理模块 A 请求鉴权模块判断 DB 客户端和 DB 服务器的标识是否合法。例如判断 DB 客户端和 DB 服务器是否具备通知权限。在合法时，例如 DB 客户端和 DB 服务器具备通知权限时，向客户机侧的第一命令队列写入第一命令，并发出第一核间中断给客户机侧。第一命令可以是 DB 客户端和 DB 服务器之间的专用数据通道建立命令，可包括 DB 客户端的标识、DB 服务器的标识、DB 客户端的阻塞信息、共享内存的参数。

在客户机侧，代理模块 B 接收第一核间中断，从第一命令队列处获取到第一命令。根据第一命令，代理模块 B 确定该命令目的是在 DB 客户端和 DB 服务器之间建立专用数据通道，代理模块 B 根据第一命令中的 DB 服务器的标识，先通过第一进程模块请求宿主机侧鉴权模块鉴权。鉴权模块判断 DB 服务器具备访问所述共享内存的权限后，通过能力安全机制授权共享内存的使用权限给 DB 服务器，建立 2 阶段页表（stage2 页表），再将 DB 服务器的权限信息经由第一进程模块返回代理模块 B。代理模块 B 在 DB 服务器具有访问共享内存的权限时，将 DB 服务器的虚拟地址也映射到共享内存上（建立 stage1 页表），从而完成专用数据通道的建立。

代理模块 B 将通道建立成功的信息写入专用数据通道，并通过写入第二命令到宿主机侧的第二命令队列、并发出第二核间中断的方式，通知宿主机侧，使宿主机侧的代理模块 A 可以从专用数据通道获取到通道建立成功的信息。代理模块 A 可解除 DB 客户端的阻塞状态。

在建立专用数据通道后，DB 客户端和 DB 服务器之间还可以进一步建立控制通道、或者建立更多的数据通道等等。同时，此专用数据通道已经经过鉴权，因此后续使用时可不必重复鉴权。

通过这种方式，使得 DB 客户端和 DB 服务器之间通信的时候，不会再发生虚拟机陷入陷出，除通道初始化过程外，客户机内核的信息收发活动可以仅限制客户机内核内部。同时在建立通道的过程中，会通过鉴权的方式保障专用数据通道的安全性，通过代理模块和应用程

序分层管理和微内核的能力安全机制，较为安全地在 DB 客户端和 DB 服务器之间开启了一条私密的专用数据通道。除 DB 客户端和 DB 服务器外的其他实体均无法通过正常手段获取到此通道的使用权限。

图 12 示出根据本申请实施例的应用程序间通信装置 10 的示例性结构示意图。

如图 12 所示，在一种可能的实现方式中，本申请实施例提出一种应用程序间通信装置 10,所述装置 10 应用于电子设备,所述电子设备包括第一虚拟装置 101 和第二虚拟装置 102，所述第一虚拟装置 101 和所述第二虚拟装置 102 中的一个是虚拟机，另一个是物理机，所述装置 10 包括：

所述第一虚拟装置 101，用于根据第一应用程序发出的第一请求，向第一命令队列中写入第一命令，所述第一请求、所述第一命令包括第二虚拟装置上的第二应用程序的标识；输出第一核间中断到所述第二虚拟装置 102 所在的核心，以使所述第二虚拟装置 102 能够获取所述第一核间中断；

所述第二虚拟装置 102，用于根据所述第一核间中断从所述第一命令队列获取所述第一命令；根据所述第一命令中、所述第二应用程序的标识运行所述第二应用程序。

在一种可能的实现方式中，所述第一虚拟装置 101 还包括第一代理模块，所述第一代理模块用于，接收所述第一应用程序发送的所述第一请求；根据所述第一请求，向所述第一命令队列中写入所述第一命令；输出所述第一核间中断到所述第二虚拟装置 102 所在的核心。

在一种可能的实现方式中，所述第二虚拟装置 102 还包括第二代理模块，所述第二代理模块用于，根据所述第一核间中断从所述第一命令队列获取所述第一命令；根据所述第一命令中、所述第二应用程序的标识运行所述第二应用程序。

在一种可能的实现方式中，所述第一请求和所述第一命令还包括所述第一应用程序的阻塞信息，所述第一代理模块接收所述第一应用程序发送的所述第一请求之后，所述第一应用程序进入阻塞状态；在接收到第二核间中断时，所述第一代理模块还用于，根据所述第二核间中断从第二命令队列获取第二命令，所述第二核间中断在所述第二应用程序运行结束后产生；在所述第二命令包括所述第一应用程序的标识时，所述第一代理模块还用于，解除所述第一应用程序的阻塞状态。

在一种可能的实现方式中，所述第二应用程序运行结束后，所述第二代理模块还用于，接收所述第二应用程序发送的第二请求，所述第二请求包括所述第一应用程序的标识；根据所述第二请求，向所述第二命令队列中写入所述第二命令，所述第二命令包括所述第一应用程序的标识；输出所述第二核间中断到所述第一虚拟装置 101 所在的核心，以使所述第一虚拟装置 101 能够获取所述第二核间中断。

在一种可能的实现方式中，所述第一代理模块根据所述第一请求，向所述第一命令队列中写入所述第一命令，包括：所述第一代理模块根据所述第一请求，判断所述第二应用程序是否具有通知权限；在所述第二应用程序具有通知权限时，向所述第一命令队列中写入所述第一命令。

在一种可能的实现方式中，所述第一虚拟装置 101 还用于响应于所述第一应用程序的申请创建共享内存，所述第一代理模块还用于，建立所述第一应用程序的地址与所述共享内存的地址的关联关系；所述第一请求和所述第一命令还包括所述共享内存的参数；在所述第二命令包括所述第一应用程序的标识时，所述第一代理模块解除所述第一应用程序的阻塞状态，

包括：在所述第二命令包括所述第一应用程序的标识以及通道建立成功的信息时，所述第一代理模块解除所述第一应用程序的阻塞状态。

在一种可能的实现方式中，所述第二代理模块根据所述第一命令中、所述第二应用程序的标识运行所述第二应用程序，包括：所述第二代理模块根据所述第一命令，判断所述第二应用程序是否具有访问所述共享内存的权限；所述第二代理模块运行所述第二应用程序，并在所述第二应用程序具有访问所述共享内存的权限时，建立所述第二应用程序的地址与所述共享内存的地址的关联关系；所述第二命令还包括通道建立成功或通道建立失败的信息。

在一种可能的实现方式中，所述共享内存还作为控制通道，所述控制通道包括第一数据位和第二数据位，所述第一数据位和所述第二数据位的状态由所述第一代理模块和所述第二代理模块控制；所述第一代理模块还用于，控制所述第一数据位处于置位状态、第二数据位处于非置位状态；所述第一代理模块输出所述第一核间中断到所述第二虚拟装置 102 所在的核心，包括：所述第一代理模块确定所述第一应用程序和所述第二应用程序之间的控制通道的第一数据位处于置位状态、第二数据位处于非置位状态时，所述第一代理模块输出第一核间中断到所述第二虚拟装置 102 所在的核心。

在一种可能的实现方式中，所述第二代理模块根据所述第一核间中断从所述第一命令队列获取所述第一命令，包括：所述第二代理模块根据所述第一核间中断，遍历所述第二虚拟装置上、所述第二代理模块对应的全部应用程序关联的控制通道；在确定所述第一应用程序和所述第二应用程序之间的控制通道的第一数据位处于置位状态时，所述第二代理模块从第一命令队列获取所述第一命令。

在一种可能的实现方式中，所述第二代理模块还用于，控制所述第一应用程序和所述第二应用程序之间的控制通道的第一数据位处于非置位状态、第二数据位处于非置位状态；针对所述第一应用程序和所述第二应用程序之间的控制通道，所述第二应用程序运行结束后，所述第二代理模块还用于，确定该控制通道的第一数据位处于非置位状态时，控制该控制通道的第二数据位处于置位状态；所述第二代理模块还用于，确定该控制通道的第一数据位处于置位状态时，重新从所述第一命令队列获取命令，并根据新获取的命令中第二应用程序的标识调度所述第二应用程序运行。

在一种可能的实现方式中，在所述第一虚拟装置 101 是物理机、所述第二虚拟装置 102 是虚拟机时，所述第一虚拟装置 101 还包括鉴权模块和第一进程模块，所述第二代理模块根据所述第一命令，判断所述第二应用程序是否具有访问所述共享内存的权限，包括：所述第二代理模块将所述第一命令中、所述第二应用程序的标识转发至所述第一虚拟装置上的第一进程模块；所述第一进程模块确定与所述第二应用程序的标识对应的鉴权标识，并向所述鉴权模块发送权限信息获取请求，所述权限信息获取请求包括所述鉴权标识；所述第一进程模块接收来自所述鉴权模块的所述第二应用程序的权限信息，并传输至所述第二代理模块；所述第二代理模块根据所述第二应用程序的权限信息，判断所述第二应用程序是否具有访问所述共享内存的权限。

图 13 示出根据本申请实施例的应用程序间通信装置的示例性结构示意图。

本申请的实施例提供了一种应用程序间通信装置，包括：处理器以及用于存储处理器可执行指令的存储器；其中，所述处理器被配置为执行所述指令时实现上述方法。

应用程序间通信装置可以设置在电子设备中，可以包括手机、可折叠电子设备、平板电

脑、桌面型计算机、膝上型计算机、手持计算机、笔记本电脑、有屏音箱、超级移动个人计算机（ultra-mobile personal computer，UMPC）、上网本、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、无人机、车载设备、智能家居设备、或智慧城市设备中的至少一种。本申请实施例对该应用程序间通信装置的具体类型不作特殊限制。

应用程序间通信装置可以包括处理器110，内部存储器121，通信模块160等。

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。例如，处理器 110 可执行本申请实施例的第一核间中断的输出等，以实现本申请实施例的应用程序间通信方法。

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。该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，例如本申请实施例中的第一命令等。如果处理器 110 需要使用该指令或数据，可从该存储器中直接调用。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。

在一些实施例中，处理器 110 可以包括一个或多个接口。接口可以包括集成电路(inter-integrated circuit，I2C)接口，通用异步收发传输器(universal asynchronous receiver/transmitter，UART)接口，通用输入输出(general-purpose input/output，GPIO)接口等。处理器 110 可以通过以上至少一种接口连接无线通信模块、显示器、摄像头等模块。

存储器 121 可以用于存储计算机可执行程序代码，该可执行程序代码包括指令。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(比如向第一命令队列中写入第一命令的应用程序等)等。存储数据区可存储应用程序间通信装置的使用过程中所创建的数据(比如第一命令等)等。此外，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，执行应用程序间通信装置的各种功能方法或数据处理。

通信模块 160 可以用于通过有线通信或者无线通信的方式接收来自其他装置或设备的数据，或者发送数据到其他装置或设备。例如可以提供应用在应用程序间通信装置上的包括WLAN(如 Wi-Fi 网络)、蓝牙(Bluetooth，BT)、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。在应用程序间通信装置连接其他装置或设备时，通信模块 160 也可以使用有线通信方案。

可以理解的是，本申请实施例示意的结构并不构成对应用程序间通信装置的具体限定。在本申请另一些实施例中，应用程序间通信装置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。

本申请的实施例提供了一种非易失性计算机可读存储介质，其上存储有计算机程序指令，所述计算机程序指令被处理器执行时实现上述方法。

本申请的实施例提供了一种计算机程序产品，包括计算机可读代码，或者承载有计算机可读代码的非易失性计算机可读存储介质，当所述计算机可读代码在电子设备的处理器中运行时，所述电子设备中的处理器执行上述方法。

计算机可读存储介质可以是可以保持和存储由指令执行设备使用的指令的有形设备。计算机可读存储介质例如可以是——但不限于——电存储设备、磁存储设备、光存储设备、电磁存储设备、半导体存储设备或者上述的任意合适的组合。计算机可读存储介质的更具体的例子(非穷举的列表)包括:便携式计算机盘、硬盘、随机存取存储器(Random Access Memory, RAM)、只读存储器（Read Only Memory，ROM)、可擦式可编程只读存储器（Electrically Programmable Read-Only-Memory，EPROM 或闪存)、静态随机存取存储器（Static Random-Access Memory，SRAM)、便携式压缩盘只读存储器(Compact Disc Read-Only Memory, CD-ROM)、数字多功能盘（Digital Video Disc，DVD)、记忆棒、软盘、机械编码设备、例如其上存储有指令的打孔卡或凹槽内凸起结构、以及上述的任意合适的组合。

这里所描述的计算机可读程序指令或代码可以从计算机可读存储介质下载到各个计算/处理设备，或者通过网络、例如因特网、局域网、广域网和/或无线网下载到外部计算机或外部存储设备。网络可以包括铜传输电缆、光纤传输、无线传输、路由器、防火墙、交换机、网关计算机和/或边缘服务器。每个计算/处理设备中的网络适配卡或者网络接口从网络接收计算机可读程序指令，并转发该计算机可读程序指令，以供存储在各个计算/处理设备中的计算机可读存储介质中。

用于执行本申请操作的计算机程序指令可以是汇编指令、指令集架构（Instruction Set Architecture，ISA）指令、机器指令、机器相关指令、微代码、固件指令、状态设置数据、或者以一种或多种编程语言的任意组合编写的源代码或目标代码，所述编程语言包括面向对象的编程语言—诸如 Smalltalk、C++等，以及常规的过程式编程语言—诸如“C”语言或类似的编程语言。计算机可读程序指令可以完全地在用户计算机上执行、部分地在用户计算机上执行、作为一个独立的软件包执行、部分在用户计算机上部分在远程计算机上执行、或者完全在远程计算机或服务器上执行。在涉及远程计算机的情形中，远程计算机可以通过任意种类的网络—包括局域网(Local Area Network，LAN)或广域网(Wide Area Network，WAN)—连接到用户计算机，或者，可以连接到外部计算机（例如利用因特网服务提供商来通过因特网连接)。在一些实施例中，通过利用计算机可读程序指令的状态信息来个性化定制电子电路，例如可编程逻辑电路、现场可编程门阵列（Field-Programmable Gate Array，FPGA）或可编程逻辑阵列（Programmable Logic Array，PLA)，该电子电路可以执行计算机可读程序指令，从而实现本申请的各个方面。

这里参照根据本申请实施例的方法、装置（系统）和计算机程序产品的流程图和/或框图描述了本申请的各个方面。应当理解，流程图和/或框图的每个方框以及流程图和/或框图中各方框的组合，都可以由计算机可读程序指令实现。

这些计算机可读程序指令可以提供给通用计算机、专用计算机或其它可编程数据处理装置的处理器，从而生产出一种机器，使得这些指令在通过计算机或其它可编程数据处理装置

的处理器执行时，产生了实现流程图和/或框图中的一个或多个方框中规定的功能/动作的装置。也可以把这些计算机可读程序指令存储在计算机可读存储介质中，这些指令使得计算机、可编程数据处理装置和/或其他设备以特定方式工作，从而，存储有指令的计算机可读介质则包括一个制造品，其包括实现流程图和/或框图中的一个或多个方框中规定的功能/动作的各个方面的指令。

也可以把计算机可读程序指令加载到计算机、其它可编程数据处理装置、或其它设备上，使得在计算机、其它可编程数据处理装置或其它设备上执行一系列操作步骤，以产生计算机实现的过程，从而使得在计算机、其它可编程数据处理装置、或其它设备上执行的指令实现流程图和/或框图中的一个或多个方框中规定的功能/动作。

附图中的流程图和框图显示了根据本申请的多个实施例的装置、系统、方法和计算机程序产品的可能实现的体系架构、功能和操作。在这点上，流程图或框图中的每个方框可以代表一个模块、程序段或指令的一部分，所述模块、程序段或指令的一部分包含一个或多个用于实现规定的逻辑功能的可执行指令。在有些作为替换的实现中，方框中所标注的功能也可以以不同于附图中所标注的顺序发生。例如，两个连续的方框实际上可以基本并行地执行，它们有时也可以按相反的顺序执行，这依所涉及的功能而定。

也要注意的是，框图和/或流程图中的每个方框、以及框图和/或流程图中的方框的组合，可以用执行相应的功能或动作的硬件（例如电路或ASIC（Application Specific Integrated Circuit，专用集成电路））来实现，或者可以用硬件和软件的组合，如固件等来实现。

尽管在此结合各实施例对本发明进行了描述，然而，在实施所要求保护的本发明过程中，本领域技术人员通过查看所述附图、公开内容、以及所附权利要求书，可理解并实现所述公开实施例的其它变化。在权利要求中，“包括”（comprising）一词不排除其他组成部分或步骤，“一”或“一个”不排除多个的情况。单个处理器或其它单元可以实现权利要求中列举的若干项功能。相互不同的从属权利要求中记载了某些措施，但这并不表示这些措施不能组合起来产生良好的效果。

以上已经描述了本申请的各实施例，上述说明是示例性的，并非穷尽性的，并且也不限于所披露的各实施例。在不偏离所说明的各实施例的范围的情况下，对于本技术领域的普通技术人员来说许多修改和变更都是显而易见的。本文中所用术语的选择，旨在最好地解释各实施例的原理、实际应用或对市场中的技术的改进，或者使本技术领域的其它普通技术人员能理解本文披露的各实施例。

权 利 要 求 书

1. 一种应用程序间通信方法，其特征在于，所述方法应用于电子设备，所述电子设备包括第一虚拟装置和第二虚拟装置，所述第一虚拟装置和所述第二虚拟装置中的一个是虚拟机，另一个是物理机，所述方法包括：

所述第一虚拟装置根据第一应用程序发出的第一请求，向第一命令队列中写入第一命令，所述第一请求、所述第一命令包括所述第二虚拟装置上的第二应用程序的标识；

所述第一虚拟装置输出第一核间中断到所述第二虚拟装置所在的核心，以使所述第二虚拟装置能够获取所述第一核间中断；

所述第二虚拟装置根据所述第一核间中断从所述第一命令队列获取所述第一命令；

所述第二虚拟装置根据所述第一命令中、所述第二应用程序的标识运行所述第二应用程序。

2. 根据权利要求 1 所述的方法，其特征在于，所述第一虚拟装置还包括第一代理模块，所述第一虚拟装置根据第一应用程序发出的第一请求，向第一命令队列中写入第一命令，包括：

所述第一代理模块接收所述第一应用程序发送的所述第一请求；

所述第一代理模块根据所述第一请求，向所述第一命令队列中写入所述第一命令；

所述第一虚拟装置输出第一核间中断到所述第二虚拟装置所在的核心，包括：

所述第一代理模块输出所述第一核间中断到所述第二虚拟装置所在的核心。

3. 根据权利要求 1 或 2 所述的方法，其特征在于，所述第二虚拟装置还包括第二代理模块，所述第二虚拟装置根据所述第一核间中断从所述第一命令队列获取所述第一命令包括：

所述第二代理模块根据所述第一核间中断从所述第一命令队列获取所述第一命令；

所述第二虚拟装置根据所述第一命令中、所述第二应用程序的标识运行所述第二应用程序包括：

所述第二代理模块根据所述第一命令中、所述第二应用程序的标识运行所述第二应用程序。

4. 根据权利要求 2 或 3 所述的方法，其特征在于，所述第一请求和所述第一命令还包括所述第一应用程序的阻塞信息，

所述第一代理模块接收所述第一应用程序发送的所述第一请求之后，所述方法还包括：

所述第一应用程序进入阻塞状态；

在接收到第二核间中断时，所述第一代理模块根据所述第二核间中断从第二命令队列获取第二命令，所述第二核间中断在所述第二应用程序运行结束后产生；

在所述第二命令包括所述第一应用程序的标识时，所述第一代理模块解除所述第一应用程序的阻塞状态。

5. 根据权利要求 4 所述的方法，其特征在于，所述第二虚拟装置根据所述第一命令中、所述第二应用程序的标识运行所述第二应用程序之后，所述方法还包括：

所述第二应用程序运行结束后，所述第二代理模块接收所述第二应用程序发送的第二请求，所述第二请求包括所述第一应用程序的标识；

所述第二代理模块根据所述第二请求，向所述第二命令队列中写入所述第二命令，所述第二命令包括所述第一应用程序的标识；

所述第二代理模块输出所述第二核间中断到所述第一虚拟装置所在的核心，以使所述第一虚拟装置能够获取所述第二核间中断。

6. 根据权利要求 2-5 中任一项所述的方法，其特征在于，所述第一代理模块根据所述第一请求，向所述第一命令队列中写入所述第一命令，包括：

所述第一代理模块根据所述第一请求，判断所述第二应用程序是否具有通知权限；

在所述第二应用程序具有通知权限时，向所述第一命令队列中写入所述第一命令。

7. 根据权利要求 4-6 中任一项所述的方法，其特征在于，所述第一虚拟装置根据所述第一应用程序发出的第一请求，向第一命令队列中写入第一命令之前，所述方法还包括：

响应于所述第一应用程序的申请，所述第一虚拟装置创建共享内存，所述第一代理模块建立所述第一应用程序的地址与所述共享内存的地址的关联关系；所述第一请求和所述第一命令还包括所述共享内存的参数；

在所述第二命令包括所述第一应用程序的标识时，所述第一代理模块解除所述第一应用程序的阻塞状态，包括：

在所述第二命令包括所述第一应用程序的标识以及通道建立成功的信息时，所述第一代理模块解除所述第一应用程序的阻塞状态。

8. 根据权利要求 7 所述的方法，其特征在于，所述第二代理模块根据所述第一命令中、所述第二应用程序的标识运行所述第二应用程序，包括：

所述第二代理模块根据所述第一命令，判断所述第二应用程序是否具有访问所述共享内

存的权限；

所述第二代理模块运行所述第二应用程序，并在所述第二应用程序具有访问所述共享内存的权限时，建立所述第二应用程序的地址与所述共享内存的地址的关联关系；所述第二命令还包括通道建立成功或通道建立失败的信息。

9. 根据权利要求 7 或 8 所述的方法，其特征在于，所述共享内存还作为控制通道，所述控制通道包括第一数据位和第二数据位，所述第一数据位和所述第二数据位的状态由所述第一代理模块和所述第二代理模块控制；

所述第一虚拟装置输出第一核间中断到所述第二虚拟装置所在的核心之前，所述方法还包括：

所述第一代理模块控制所述第一数据位处于置位状态、第二数据位处于非置位状态；

所述第一代理模块输出所述第一核间中断到所述第二虚拟装置所在的核心，包括：

所述第一代理模块确定所述第一应用程序和所述第二应用程序之间的控制通道的第一数据位处于置位状态、第二数据位处于非置位状态时，所述第一代理模块输出第一核间中断到所述第二虚拟装置所在的核心。

10. 根据权利要求 9 所述的方法，其特征在于，所述第二代理模块根据所述第一核间中断从所述第一命令队列获取所述第一命令，包括：

所述第二代理模块根据所述第一核间中断，遍历所述第二虚拟装置上、所述第二代理模块对应的全部应用程序关联的控制通道；

在确定所述第一应用程序和所述第二应用程序之间的控制通道的第一数据位处于置位状态时，所述第二代理模块从第一命令队列获取所述第一命令。

11. 根据权利要求 9 或 10 所述的方法，其特征在于，所述第二虚拟装置根据所述第一命令中、所述第二应用程序的标识运行所述第二应用程序之后，所述方法还包括：

所述第二代理模块控制所述第一应用程序和所述第二应用程序之间的控制通道的第一数据位处于非置位状态、第二数据位处于非置位状态；

针对所述第一应用程序和所述第二应用程序之间的控制通道，所述第二应用程序运行结束后，所述第二代理模块确定该控制通道的第一数据位处于非置位状态时，控制该控制通道的第二数据位处于置位状态；

所述第二代理模块确定该控制通道的第一数据位处于置位状态时，重新从所述第一命令队列获取命令，并根据新获取的命令中第二应用程序的标识调度所述第二应用程序运行。

12. 根据权利要求 6-11 中任一项所述的方法，其特征在于，在所述第一虚拟装置是物理

机、所述第二虚拟装置是虚拟机时，所述第一虚拟装置还包括鉴权模块和第一进程模块，所述第二代理模块根据所述第一命令，判断所述第二应用程序是否具有访问所述共享内存的权限，包括：

所述第二代理模块将所述第一命令中、所述第二应用程序的标识转发至所述第一虚拟装置上的第一进程模块；

所述第一进程模块确定与所述第二应用程序的标识对应的鉴权标识，并向所述鉴权模块发送权限信息获取请求，所述权限信息获取请求包括所述鉴权标识；

所述第一进程模块接收来自所述鉴权模块的所述第二应用程序的权限信息，并传输至所述第二代理模块；

所述第二代理模块根据所述第二应用程序的权限信息，判断所述第二应用程序是否具有访问所述共享内存的权限。

13. 一种应用程序间通信装置，其特征在于，所述装置应用于电子设备，所述电子设备运行第一虚拟装置和第二虚拟装置，所述第一虚拟装置和所述第二虚拟装置中的一个是虚拟机，另一个是物理机，所述装置包括：

所述第一虚拟装置，用于根据第一应用程序发出的第一请求，向第一命令队列中写入第一命令，所述第一请求、所述第一命令包括第二虚拟装置上的第二应用程序的标识；输出第一核间中断到所述第二虚拟装置所在的核心，以使所述第二虚拟装置能够获取所述第一核间中断；

所述第二虚拟装置，用于根据所述第一核间中断从所述第一命令队列获取所述第一命令；根据所述第一命令中、所述第二应用程序的标识运行所述第二应用程序。

14. 一种应用程序间通信装置，其特征在于，包括：

处理器；

用于存储处理器可执行指令的存储器；

其中，所述处理器被配置为执行所述指令时实现权利要求 1-12 任意一项所述的方法。

15. 一种非易失性计算机可读存储介质，其上存储有计算机程序指令，其特征在于，所述计算机程序指令被处理器执行时实现权利要求 1-12 中任意一项所述的方法。

16. 一种计算机程序产品，包括计算机可读代码，或者承载有计算机可读代码的非易失性计算机可读存储介质，其特征在于，当所述计算机可读代码在电子设备中运行时，所述电子设备中的处理器执行权利要求 1-12 中任意一项所述的方法。

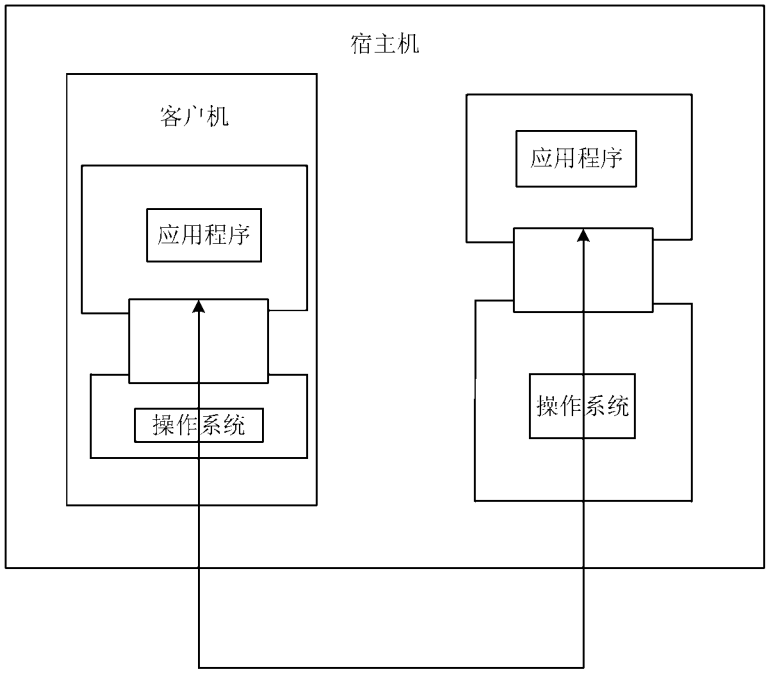

宿主机
客户机
应用程序
操作系统
应用程序
操作系统

图 1

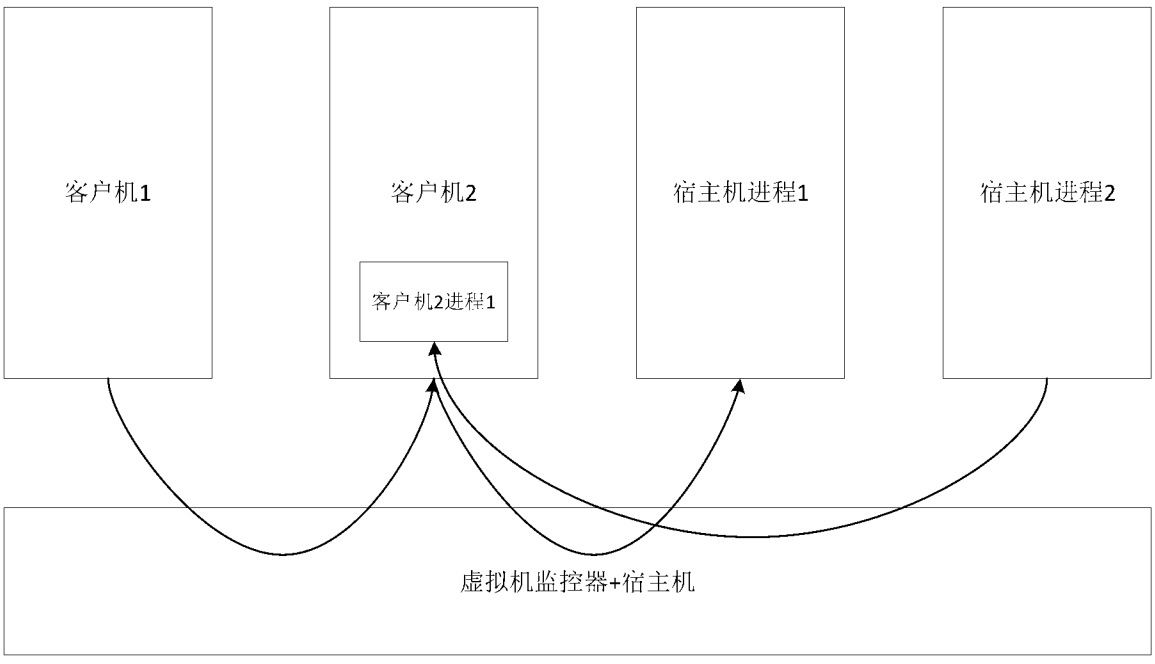

客户机1
客户机2
客户机2进程1
宿主机进程1
宿主机进程2
虚拟机监控器+宿主机

图 2

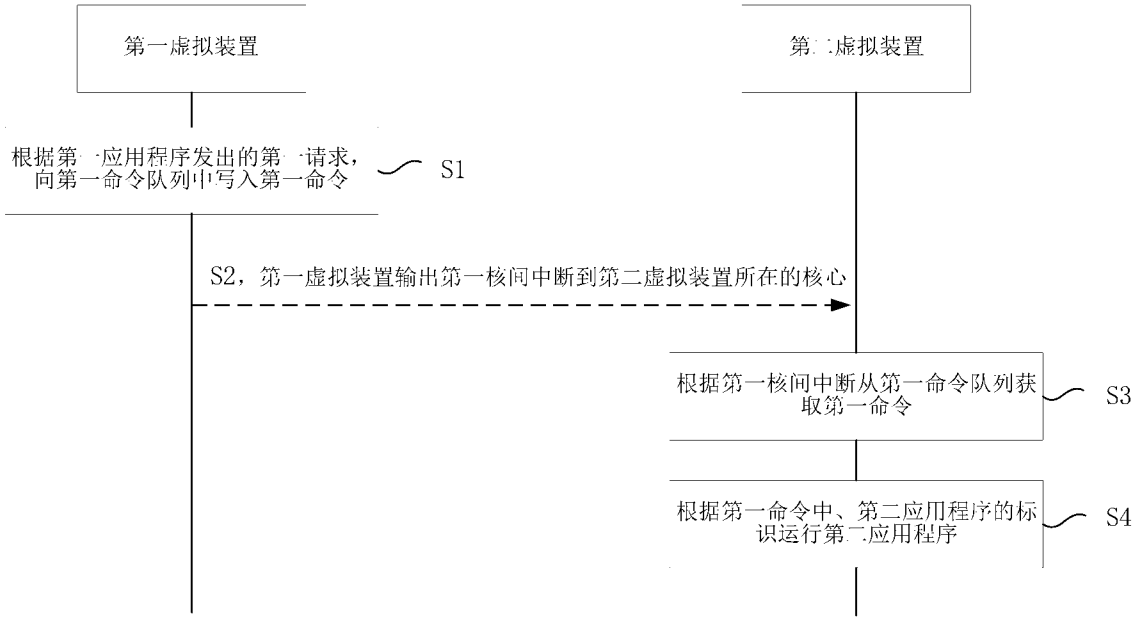

第一虚拟装置
第二虚拟装置
根据第一应用程序发出的第一请求，向第一命令队列中写入第一命令
S1
S2，第一虚拟装置输出第一核间中断到第二虚拟装置所在的核心
根据第一核间中断从第一命令队列获取第一命令
S3
根据第一命令中、第二应用程序的标识运行第二应用程序
S4

图 3

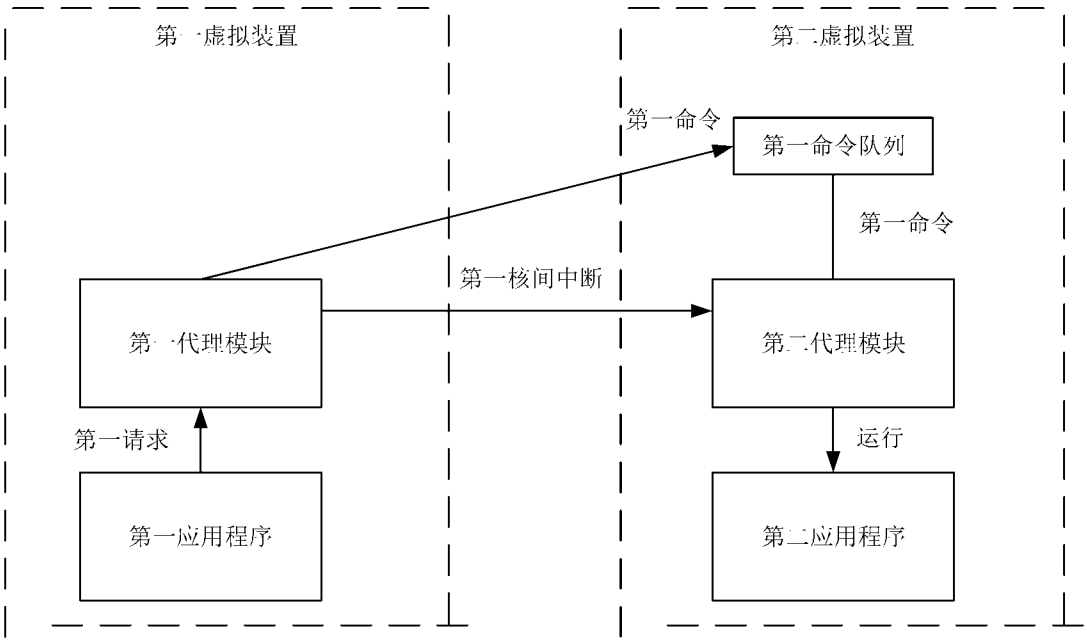

第一虚拟装置
第二虚拟装置
第一命令
第一命令队列
第一命令
第一核间中断
第一代理模块
第二代理模块
第一请求
运行
第一应用程序
第二应用程序

图 4

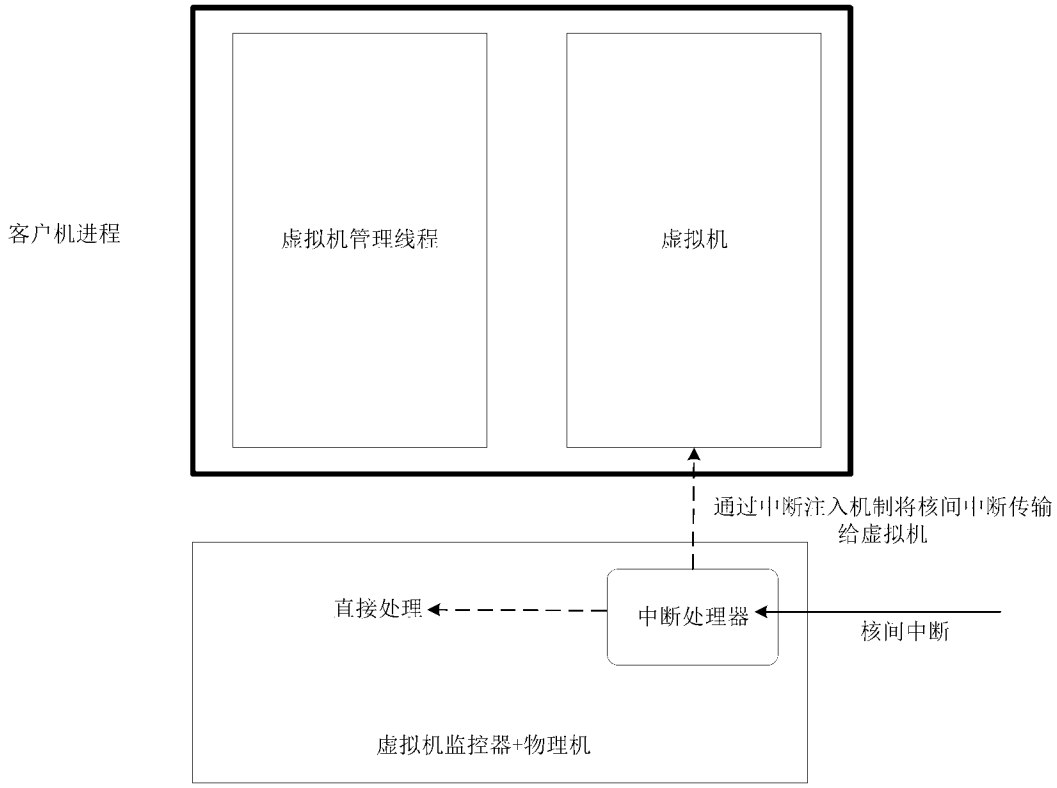

客户机进程
虚拟机管理线程
虚拟机
通过中断注入机制将核间中断传输给虚拟机
直接处理
中断处理器
核间中断
虚拟机监控器+物理机

图 5a

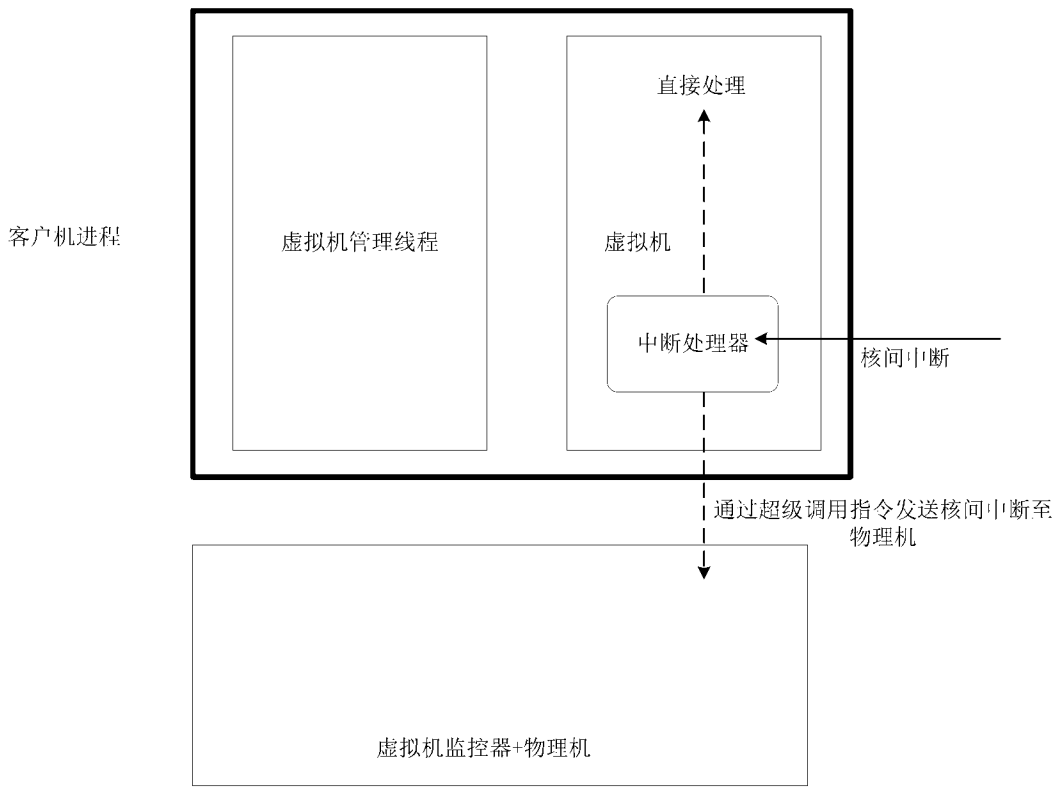

客户机进程
虚拟机管理线程
直接处理
虚拟机
中断处理器
核间中断
通过超级调用指令发送核间中断至物理机
虚拟机监控器+物理机

图 5b

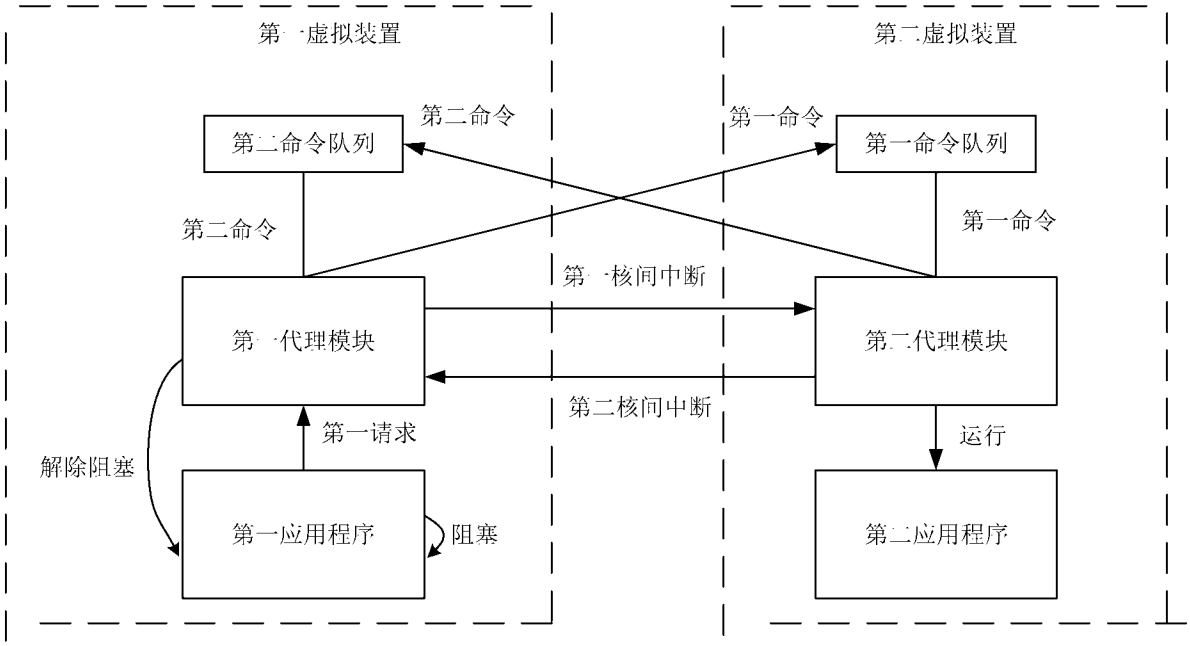

第一虚拟装置
第二虚拟装置
第二命令队列
第二命令
第一命令
第一命令队列
第二命令
第一命令
第一核间中断
第一代理模块
第二代理模块
第二核间中断
第一请求
运行
解除阻塞
第一应用程序
阻塞
第二应用程序

图 6

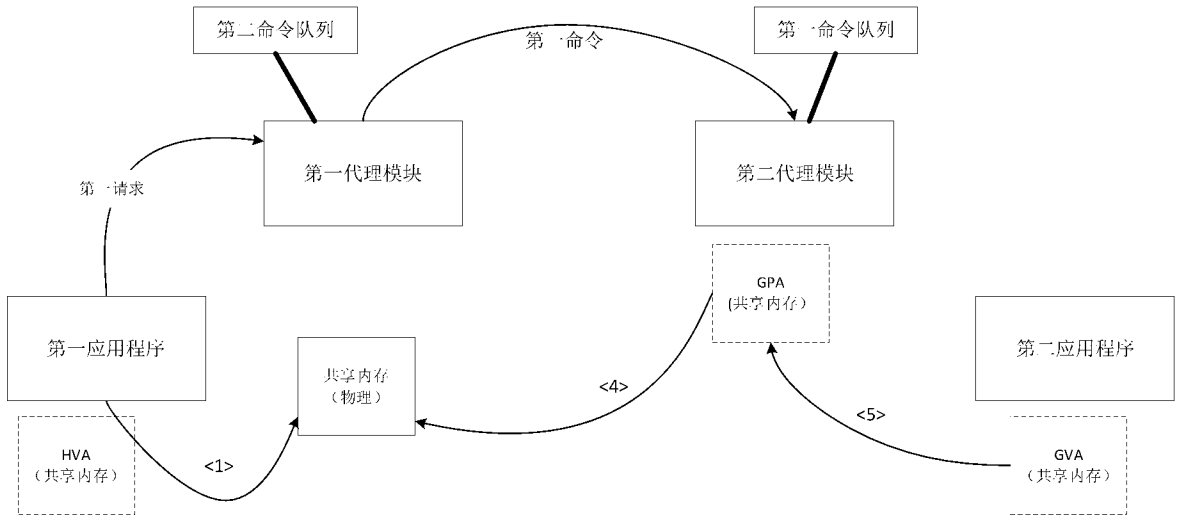

第二命令队列
第一命令
第一命令队列
第一请求
第一代理模块
第二代理模块
GPA
(共享内存)
第一应用程序
共享内存
(物理)
第二应用程序
<4>
<5>
<1>
HVA
(共享内存)
GVA
(共享内存)

图 7

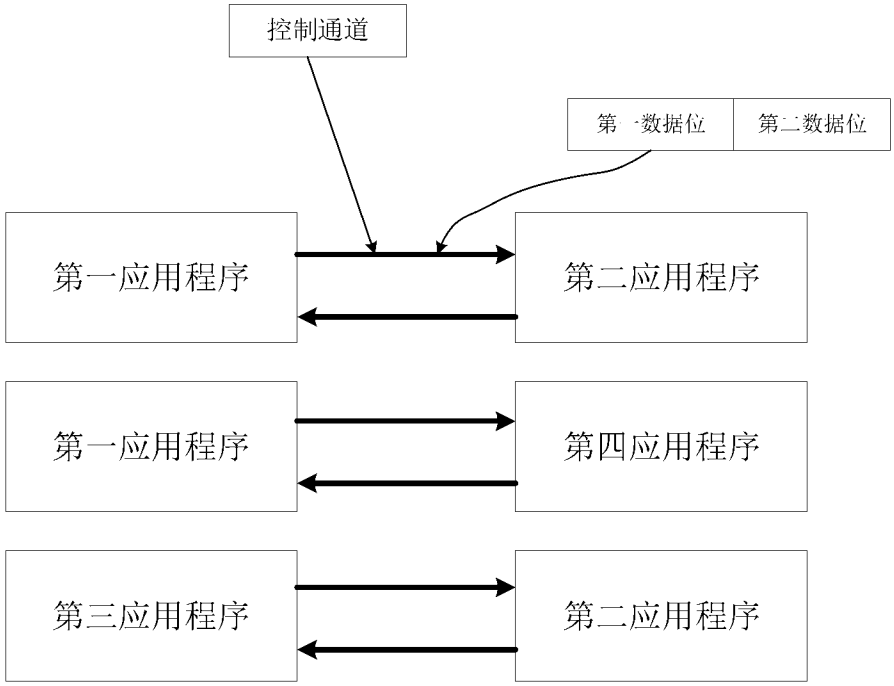

控制通道
第一数据位
第二数据位
第一应用程序
第二应用程序
第一应用程序
第四应用程序
第三应用程序
第二应用程序

图 8

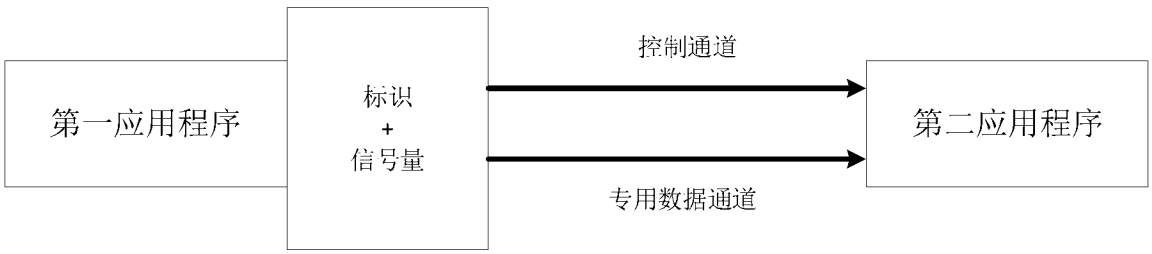

第一应用程序
标识
+
信号量
控制通道
专用数据通道
第二应用程序

图 9

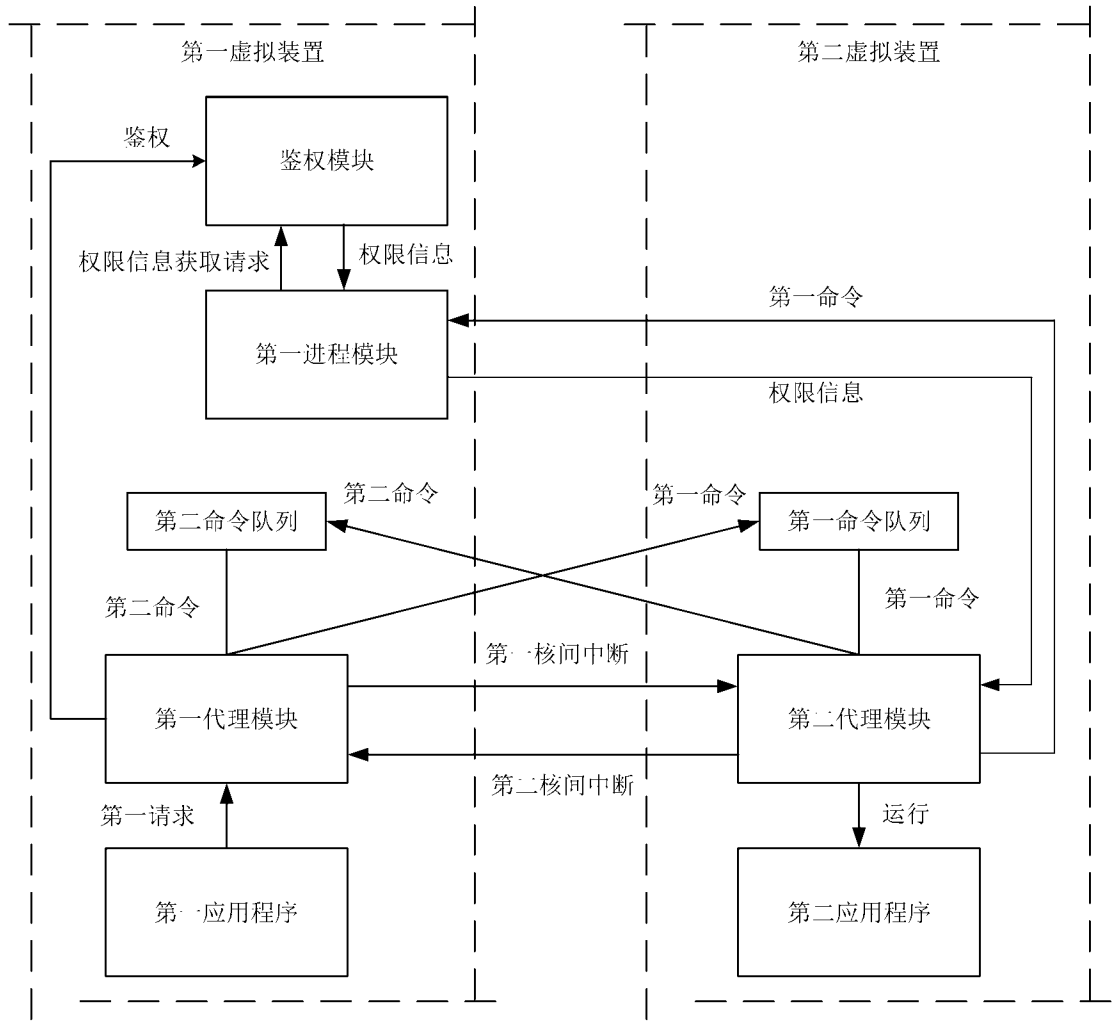

第一虚拟装置
第二虚拟装置
鉴权
鉴权模块
权限信息获取请求
权限信息
第一命令
第一进程模块
权限信息
第二命令
第一命令
第二命令队列
第一命令队列
第二命令
第一命令
第一核间中断
第一代理模块
第二代理模块
第二核间中断
第一请求
运行
第一应用程序
第二应用程序

图 10a

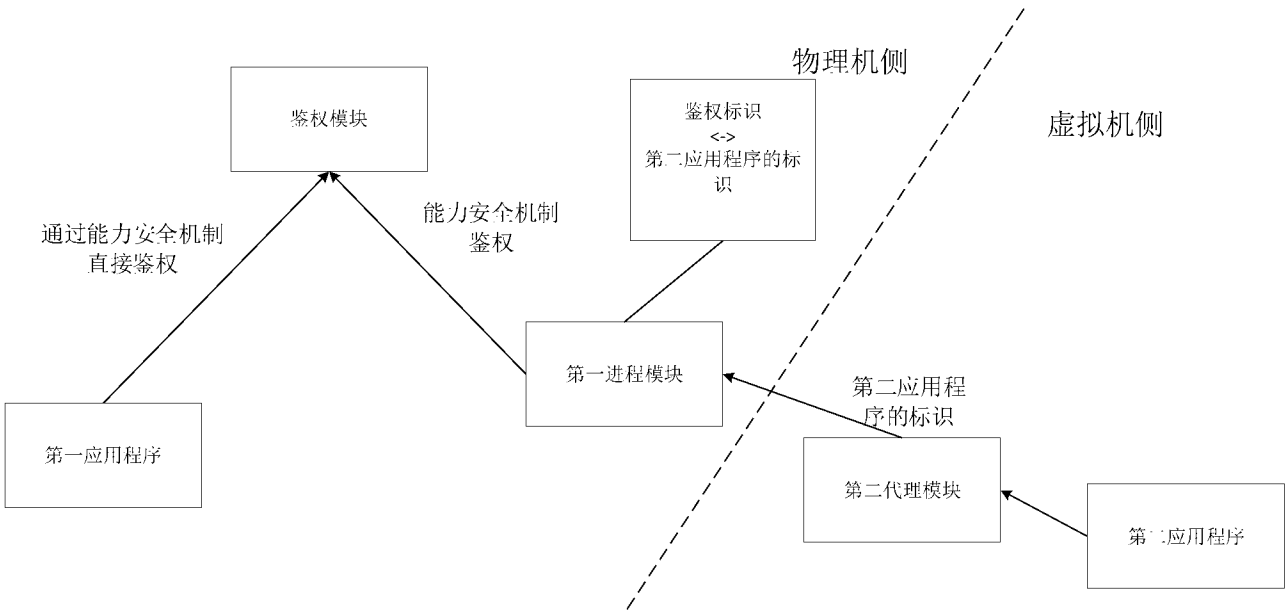

物理机侧
虚拟机侧
鉴权模块
鉴权标识
<->
第二应用程序的标识
通过能力安全机制
直接鉴权
能力安全机制
鉴权
第一进程模块
第二应用程
序的标识
第一应用程序
第二代理模块
第二应用程序

图 10b

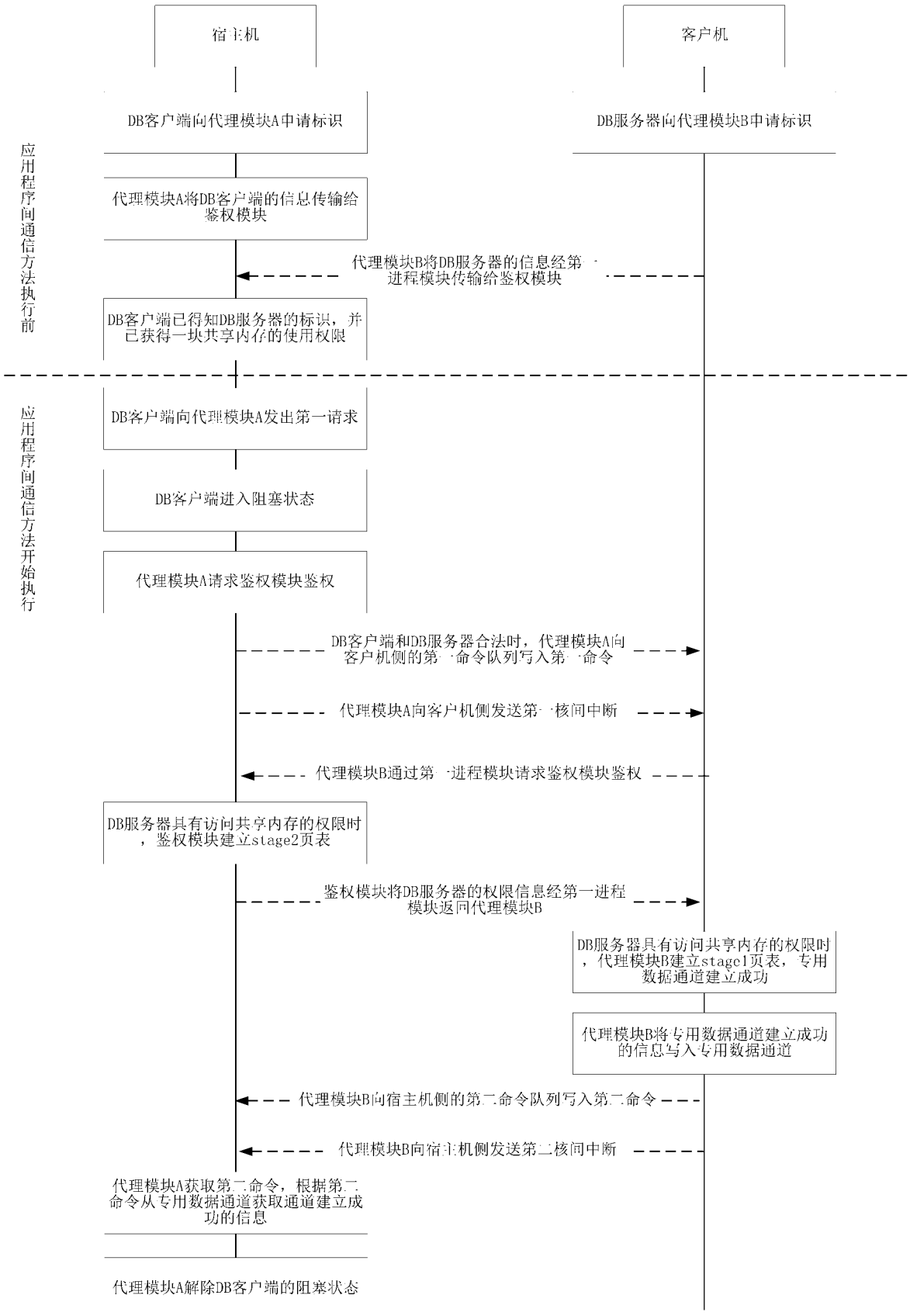

宿主机
客户机
应用程序间通信方法执行前
DB客户端向代理模块A申请标识
DB服务器向代理模块B申请标识
代理模块A将DB客户端的信息传输给鉴权模块
代理模块B将DB服务器的信息经第一进程模块传输给鉴权模块
DB客户端已得知DB服务器的标识，并已获得一块共享内存的使用权限
应用程序间通信方法开始执行
DB客户端向代理模块A发出第一请求
DB客户端进入阻塞状态
代理模块A请求鉴权模块鉴权
DB客户端和DB服务器合法时，代理模块A向客户机侧的第一命令队列写入第一命令
代理模块A向客户机侧发送第一核间中断
代理模块B通过第一进程模块请求鉴权模块鉴权
DB服务器具有访问共享内存的权限时，鉴权模块建立stage2页表
鉴权模块将DB服务器的权限信息经第一进程模块返回代理模块B
DB服务器具有访问共享内存的权限时，代理模块B建立stage1页表，专用数据通道建立成功
代理模块B将专用数据通道建立成功的信息写入专用数据通道
代理模块B向宿主机侧的第二命令队列写入第二命令
代理模块B向宿主机侧发送第二核间中断
代理模块A获取第二命令，根据第二命令从专用数据通道获取通道建立成功的信息
代理模块A解除DB客户端的阻塞状态

图 11

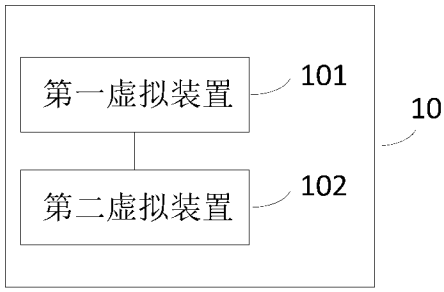


图 12

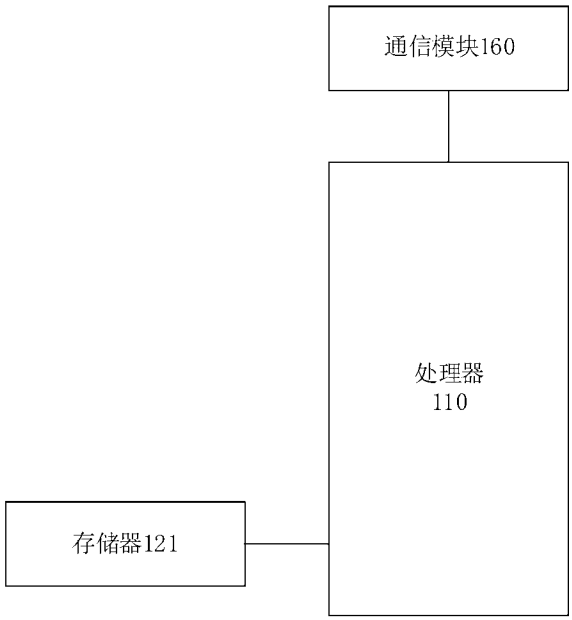


图 13

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2023/081645

A. CLASSIFICATION OF SUBJECT MATTER

G06F 9/54(2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC: G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

VEN, CNABS, CNTXT, WOTXT, EPTXT, USTXT, CNKI, IEEE: 核间中断, 客户机, 宿主机, 通信, 物理机, 虚拟机, 应用程序, interrupt, guest, host, communication, physical, machine, virtual, application

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	CN 113778611 A (RUIJIE NETWORKS (SUZHOU) CO., LTD.) 10 December 2021 (2021-12-10) description, paragraphs [0045]-[0066]	1-16
A	CN 102693162 A (SUZHOU INSTITUTE FOR ADVANCED STUDY, USTC) 26 September 2012 (2012-09-26) entire document	1-16
A	CN 108304269 A (ALIBABA GROUP HOLDING LTD.) 20 July 2018 (2018-07-20) entire document	1-16
A	CN 108845863 A (BEIJING BAIDU NETCOM SCIENCE AND TECHNOLOGY CO., LTD.) 20 November 2018 (2018-11-20) entire document	1-16
A	US 2015082300 A1 (NETAPP. INC.) 19 March 2015 (2015-03-19) entire document	1-16

☐ Further documents are listed in the continuation of Box C. ☑ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"D" document cited by the applicant in the international application
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
04 June 2023	**09 June 2023**

Name and mailing address of the ISA/CN	Authorized officer
China National Intellectual Property Administration (ISA/CN) **China No. 6, Xitucheng Road, Jimenqiao, Haidian District, Beijing 100088**	Telephone No.

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2023/081645

Patent document cited in search report	Publication date (day/month/year)	Patent family member(s)	Publication date (day/month/year)
CN 113778611 A	10 December 2021	None	
CN 102693162 A	26 September 2012	None	
CN 108304269 A	20 July 2018	None	
CN 108845863 A	20 November 2018	US 2019377593 A1	12 December 2019
US 2015082300 A1	19 March 2015	None	

国际检索报告

国际申请号
PCT/CN2023/081645

A. 主题的分类

G06F 9/54(2006.01)i

按照国际专利分类(IPC)或者同时按照国家分类和IPC两种分类

B. 检索领域

检索的最低限度文献(标明分类系统和分类号)

IPC：G06F

包含在检索领域中的除最低限度文献以外的检索文献

在国际检索时查阅的电子数据库(数据库的名称，和使用的检索词（如使用））

VEN,CNABS,CNTXT,WOTXT,EPTXT,USTXT,CNKI,IEEE:核间中断，客户机，宿主机，通信，物理机，虚拟机，应用程序，interrupt，guest，host，communication， physical，machine，virtual，application

C. 相关文件

类 型*	引用文件，必要时，指明相关段落	相关的权利要求
A	CN 113778611 A (锐捷网络（苏州）有限公司) 2021年12月10日 (2021-12-10) 说明书第[0045]-[0066]段	1-16
A	CN 102693162 A (中国科学技术大学苏州研究院) 2012年9月26日 (2012-09-26) 全文	1-16
A	CN 108304269 A (阿里巴巴集团控股有限公司) 2018年7月20日 (2018-07-20) 全文	1-16
A	CN 108845863 A (北京百度网讯科技有限公司) 2018年11月20日 (2018-11-20) 全文	1-16
A	US 2015082300 A1 (NETAPP. INC.) 2015年3月19日 (2015-03-19) 全文	1-16

☐ 其余文件在C栏的续页中列出。

☑ 见同族专利附件。

* 引用文件的具体类型：

“A” 认为不特别相关的表示了现有技术一般状态的文件

“D” 申请人在国际申请中引证的文件

“E” 在国际申请日的当天或之后公布的在先申请或专利

“L” 可能对优先权要求构成怀疑的文件，或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件（如具体说明的）

“O” 涉及口头公开、使用、展览或其他方式公开的文件

“P” 公布日先于国际申请日但迟于所要求的优先权日的文件

“T” 在申请日或优先权日之后公布，与申请不相抵触，但为了理解发明之理论或原理的在后文件

“X” 特别相关的文件，单独考虑该文件，认定要求保护的发明不是新颖的或不具有创造性

“Y” 特别相关的文件，当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时，要求保护的发明不具有创造性

“&” 同族专利的文件

国际检索实际完成的日期	国际检索报告邮寄日期
2023年6月4日	2023年6月9日

ISA/CN的名称和邮寄地址	受权官员
中国国家知识产权局 中国北京市海淀区蓟门桥西土城路6号 100088	孙国辉 电话号码 (+86) 010-53961538

国际检索报告
关于同族专利的信息

国际申请号
PCT/CN2023/081645

检索报告引用的专利文件	公布日 (年/月/日)	同族专利	公布日 (年/月/日)
CN 113778611 A	2021年12月10日	无	
CN 102693162 A	2012年9月26日	无	
CN 108304269 A	2018年7月20日	无	
CN 108845863 A	2018年11月20日	US 2019377593 A1	2019年12月12日
US 2015082300 A1	2015年3月19日	无	