(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 107305509 B
(45)授权公告日 2023.07.04

(21)申请号 201710160628.1
(22)申请日 2017.03.17
(65)同一申请的已公布的文献号
申请公布号 CN 107305509 A
(43)申请公布日 2017.10.31
(30)优先权数据
62/326,537 2016.04.22 US
15/333,010 2016.10.24 US
(73)专利权人 三星电子株式会社
地址 韩国京畿道水原市
(72)发明人 马诺吉·K·哥达拉
文卡塔·布哈努·普拉卡斯·格拉普提
(74)专利代理机构 北京铭硕知识产权代理有限公司 11286
专利代理师 曾世骁 于硕
(51)Int.Cl.
G06F 9/54(2006.01)
G06F 13/28(2006.01)
(56)对比文件
CN 101017461 A,2007.08.15
CN 102402487 A,2012.04.04
US 8015388 B1,2011.09.06
王庆民;刘福岩;连嘉.ARM对SDSM操作系统虚地址转换支持研究.微计算机信息.2007,(第11期),第170-172页.
审查员 张甜

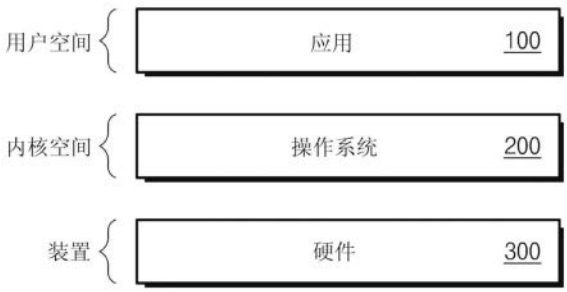
权利要求书2页 说明书4页 附图4页

(54)发明名称

涉及内存的预分配的缓冲区映射方案

(57)摘要

本发明构思涉及内存的预分配的缓冲区映射方案,涉及一种应用、操作系统和硬件互相通信的计算机实现方法。所述方法涉及所述应用将应用层虚拟地址转换为物理地址并将物理地址传送到操作系统。所述操作系统随后使用物理地址确定OS层虚拟地址并完成数据传送。



CN 107305509 B

1.一种操作系统与第一应用和第二应用进行通信的通过计算机实现的方法,所述方法包括:

直接从第一应用接收与第一应用对应的内存的第一物理地址,其中,第一物理地址通过使用第一物理页帧号和第一偏移而被确定,并且第一物理页帧号通过使用第一应用的第一页表将第一应用的虚拟页的虚拟页帧号映射到内存的物理页帧号而被确定;

使用应用偏移使第二应用与第一应用进行通信,其中,应用偏移被施加到第一物理地址,以确定与第二应用对应的第二物理地址;

基于第一物理地址确定操作系统层虚拟地址以实现数据传送,

其中,操作系统层虚拟地址通过内核转换表基于第一物理地址被确定,并且操作系统通过第一物理地址与的第一应用和第二应用通信。

2.如权利要求1所述的方法,还包括:

在第一应用或第二应用接收用户输入之前执行内存分配并与第一应用和第二应用共享所述内存分配。

3.如权利要求1所述的方法,其中,在操作系统中存在多个模块,每个模块具有它自己的操作系统层虚拟内存,

所述方法还包括:使所有的模块使用第一物理地址直接与第一应用进行通信。

4.一种在单个节点中的第一应用、第二应用、操作系统和硬件使用缓冲区互相通信的通过计算机实现的方法,所述方法包括:

第一应用将第一应用层虚拟地址转换为与第一应用对应的第一物理地址并将第一物理地址传送给所述操作系统;

第一应用和第二应用使用应用偏移进行互相通信,其中,应用偏移被施加到第一物理地址,以确定与第二应用对应的第二物理地址;

所述操作系统使用第一物理地址以确定操作系统层虚拟地址,

其中,操作系统层虚拟地址通过内核转换表基于第一物理地址被确定,并且所述操作系统通过第一物理地址与第一应用和第二应用通信。

5.如权利要求4所述的方法,其中,第一应用在不涉及所述操作系统的情况下使用利用偏移计算出的直接内存访问地址与所述硬件进行通信。

6.如权利要求4所述的方法,其中,所述操作系统在接收用户输入之前预分配内存缓冲区并向应用提供用于映射物理地址的方法。

7.如权利要求4所述的方法,其中,第一应用在通过使用缓冲区在第一应用、第二应用、所述操作系统和所述硬件之间传送数据之前获得第一物理地址和内存的直接内存访问地址。

8.如权利要求4所述的方法,其中,在所述操作系统中存在多个模块,其中,每个模块直接与第一应用进行通信并使用第一物理地址来确定它的操作系统层虚拟地址。

9.一种用于控制数据传送的设备,所述设备包括:

内存映射器,使第一应用将第一应用层虚拟地址转换为与第一应用对应的第一物理地址并将第一物理地址传送给操作系统,并且使第二应用使用应用偏移与第一应用进行通信,其中,应用偏移被施加到第一物理地址,以确定与第二应用对应的第二物理地址,

其中,所述操作系统使用第一物理地址以确定操作系统层虚拟地址,

其中，操作系统层虚拟地址通过内核转换表基于第一物理地址被确定，并且所述操作系统通过第一物理地址与第一应用和第二应用通信。

10.如权利要求9所述的设备，其中，内存映射器使第一应用使用直接内存访问地址和偏移直接与硬件进行通信。

涉及内存的预分配的缓冲区映射方案

[0001] 本申请要求于2016年4月22日提交的62/326,537号美国临时申请和于2016年10月24日提交的15/333,010号美国专利申请的权益,该申请通过引用合并于此。

技术领域

[0002] 本公开通常涉及内存缓冲区,并更具体地涉及一种涉及内存的预分配的缓冲区映射方案。

背景技术

[0003] 在基于UNIX的存储器/服务器系统中,存在各种应用和装置驱动器,每个应用和装置驱动程序执行特定的任务。为了使应用、操作系统(OS)/内核以及硬件有效地通信,它们经常分发内存缓冲区。典型地,在这些通信期间,应用将它的应用层虚拟地址传送到操作系统/内核。内存缓冲区使用应用层虚拟地址调用驱动程序,并且驱动程序将应用层虚拟地址映射到操作系统/内核层虚拟地址。

[0004] 为了使这些转换变得更简单,虚拟内存和物理内存被划分为小型的块,其中,所述小型的块被称为页。在这种分页模型中,虚拟地址由偏移和虚拟页帧号组成。每次处理器遇到虚拟地址时,处理器从虚拟地址中提取偏移和虚拟页帧号。处理器随后将虚拟页帧号转换为物理页帧号以进入该物理页访问在该正确偏移处的位置。为了将虚拟地址转换为物理地址,处理器首先算出该虚拟页内的虚拟地址页帧号和偏移。处理器使用虚拟页帧数作为进入所处理的页表内的索引以检索它的页表项。如果在该偏移处的页表项是有效的,则处理器从该项取得物理页帧号。处理器用来将虚拟页帧号转换为物理页帧号的表被称为页表。

[0005] 通过将偏移与虚拟页号相加来计算虚拟地址。为了更进一步实施保护,存在针对用户空间应用和内核的单独的页表。为了访问用户空间虚拟地址,内核层软件将用户空间地址映射到内核地址空间。这个处理涉及针对用户空间地址创建内核页表项。

[0006] 关于硬件,OS/内核与硬件之间的连接通过直接内存访问(DMA)方式发生。通过使用DMA,硬件装置可在不涉及CPU的情况下将数据传送到计算机的主内存/从计算机的主内存传送数据。为了DMA工作,内存缓冲区被频繁地映射到硬件装置可见的地址范围。这个地址范围被称作IO虚拟地址。根据架构,这种方式涉及建立IO虚拟地址与计算机主内存的物理地址之间的转换。这通常使用IOMMU硬件。对于一些架构,IO虚拟地址可与计算机主内存的物理地址相同。

[0007] 上述的映射方案给OS/内核带来沉重的负担,其中,OS/内核需要首先通过设置页表项将应用层虚拟地址转换为OS层虚拟地址。同样地,应针对每个DMA传送确定DMA映射。需要一种更有效的用于OS、应用和硬件通信的方法。

发明内容

[0008] 在一个方面,本发明构思涉及操作系统与应用进行通信的计算机实现的方法。所

述方法涉及操作系统直接从应用接收物理地址，并基于内存的物理地址确定OS层虚拟地址。

[0009] 在另一方面，本发明构思属于应用、操作系统和硬件在单个节点中与另一节点进行通信的计算机实现的方法。所述方法涉及应用将应用层虚拟地址转换为物理地址并将物理地址传送到操作系统。操作系统随后使用物理地址确定OS层虚拟地址。

[0010] 在另一方面，本发明构思涉及用于控制数据传送的设备，所述设备包括允许应用将它的应用层虚拟地址转换为物理地址并将物理地址传送到操作系统的内存映射器。

附图说明

[0011] 图1是在根据一个实施例的提供一种可能的环境的单个节点中的用户空间、内核空间和硬件的概念示图。

[0012] 图2A是示出在根据一个实施例的应用、操作系统和硬件之间的通信的示意图。

[0013] 图2B是示出根据一个实施例的用户空间中的多重应用以及使缓冲区能够被共享的内核中的指向同一物理地址的不同的虚拟地址的另一示意图。

[0014] 图3是示出根据一个实施例的在应用、操作系统和硬件之间的通信方法的示图。

具体实施方式

[0015] 本系统通过每次通过内存缓冲区时必须设置基于内核层页表的转换来节省OS。在本公开，应用将物理地址传送到内核。根据一个实施例，内核具有针对该缓冲区需要的映射。因此，内核可计算虚拟地址并不需要每次执行映射操作。因为所有的内核模块共享同一虚拟地址空间，任何OS模块(不只是被分配内存的OS模块)可使用物理地址获得虚拟地址并在缓冲区进行操作。

[0016] 在不同的应用之间，使用缓冲区偏移来进行通信。应用使用虚拟地址在缓冲区进行操作。应用可通过简单地将偏移与缓冲区起点的虚拟地址相加来计算它自己的虚拟地址。

[0017] 应用可通过简单地将偏移与缓冲区起点处的DMA地址相加来确定偏移的DMA地址。应用可在不涉及内核的情况下直接将缓冲区地址传送给硬件。

[0018] 虽然在单个节点的情况下描述本公开，但是这不是对本公开的限制。

[0019] 图1是在根据一个实施例的提供一种可能的环境的单个节点中的用户空间、内核空间和硬件的概念示图。如图所示，形成用户空间的应用100、操作系统(OS)/内核200和硬件300互相通信以接收并执行用户请求。硬件300包括各种装置、中央处理器和系统内存。操作系统200除了进行其它处理之外还在用户空间和硬件300之间进行连接，并允许应用100访问系统内存。装置驱动程序通常是OS 200的一部分。内存映射器将图像和数据文件映射到用户空间中的应用中。在内存映射中，文件的内容被链接到虚拟地址。

[0020] 图2A是示出在根据一个实施例的应用、操作系统和硬件之间的通信的示意图。关于应用100，示出了两个应用，应用X和应用Y。两个应用中的每一个具有其自己的虚拟内存，其中，其自己的虚拟内存具有其自己的一组虚拟地址，如图2A中VM-X和VN-Y所示。每个应用也具有其自己的页表110，其中，其自己的页表110将其自己的各个虚拟页映射到内存的物理页。例如，如图所示，应用X的虚拟页帧号0(VPFN 0)被映射到物理页帧号1(PFN 1)中的内

存,并且应用Y的虚拟页帧号1(VPFN 1)被映射到物理页帧号4(PFN 4)中。

[0021] 使用虚拟页帧号作为偏移访问页表110。为了将虚拟地址转换为物理地址,首先确定虚拟页中的虚拟地址页帧号和偏移。如果虚拟内存地址是有效的并且表项是有效的,则处理器获取物理页帧号并将它乘以页大小以获得物理内存中的页的底部的地址。随后,与偏移相加。

[0022] 例如,在图2A示出的情况中,假定页大小为0x2000。针对VM-Y中的地址0x2194,处理器将该地址转换为虚拟页帧号1中的偏移0x194。该虚拟页帧号1被映射为物理页帧号4,其中,物理页帧号4起始于0x8000(4x2000)。将0x194偏移与物理页帧号相加产生最终的物理地址0x8194。当应用仅使用虚拟地址和距离底部的虚拟地址的偏移彼此通信时,本系统允许应用使用物理地址与内核进行通信。如图所示,内核转换表210被用于将物理地址转换为OS层虚拟地址。内核转换表210允许从物理到虚拟地址的转换,并可以是特定OS。

[0023] 根据一实施例,内存被预分配并与应用100共享,使得应用100和操作系统200两者可访问物理地址表。这里使用的“预分配”表示在缓冲区的任何使用之前进行分配以在应用/内核/硬件域之间传送数据。此外,操作系统200中不同的模块,将物理地址转换为它自己的OS层虚拟地址,其中,所有的模块共享同一虚拟地址空间。每个OS用来将物理地址转换为虚拟地址的方法取决于每个OS的结构。例如,Linux OS可使用针对地址的特定范围的简单算术来将物理地址转换为虚拟地址。当在Linux中实施时,本系统的预分配的缓存区落在使用简单算术以到达物理地址的地址范围中。另一些OS可使用不同的机制进行该操作。

[0024] 应用可通过简单地将偏移与缓冲区起点的DMA地址相加来计算偏移的DMA地址。在这种方式中,应用可在不涉及操作系统200的情况下直接地将缓冲区地址传送到硬件300。

[0025] 图2B是示出根据一个实施例的用户空间中的多个应用以及使缓冲区能够共享的内核中指向同一物理地址的不同的虚拟地址的另一示意图。图2B示出应用用户空间100中的应用X和应用Y。在应用X中,被标为“缓冲区-1”的数据被存储在应用X地址空间中的0x3000处。例如,该数据通过使用图2A中描述的处理向内核地址空间0x5000转换。相同的数据(缓冲区-1)与在应用Y地址空间中的地址0x1000处的数据相应,但是应用X和Y两者都能够使用物理地址指向相同的数据。被标为“缓冲区-2”的数据被存储在应用Y的虚拟地址0x4000处,其中,应用Y的虚拟地址0x4000与内核地址空间0x7000相应。由于内核模块共享相同虚拟地址空间的事实,任何OS模块可使用图2B示出的内核地址获得虚拟地址。

[0026] 图3是示出根据一个实施例的在应用、操作系统和硬件之间的通信方法的示图。更具体地,图3的实施例示出用户空间(应用100)中的应用-1 102和应用-2 104通过使用偏移106彼此通信,并且应用-2 104通过使用物理地址204与内核模块202进行通信。应用(在本示例中,应用-2 104)也可使用通过使用偏移产生的DMA地址来与硬件装置302直接进行通信。

[0027] 根据一个实施例,本系统包括机器可读存储器,在其上存储了具有机器可执行的至少一个代码段的计算机程序,从而使机器执行上述步骤。

[0028] 根据一个实施例,本系统可以以硬件、软件或硬件和软件组合的形式实现。虽然本公开主要关注于涉及一个计算机程序的单节点实现,但是其也可适用于不同的元件分布在若干互相连接的计算机系统的分布式方式。用于实施这里描述的方法的任何种类的计算机系统或设备是合适的。硬件和软件的典型的组合可以是具有当被加载并执行时控制计算机

系统使得计算机系统实现这里描述的方法的计算机程序的通用计算机系统。

[0029] 本系统可被嵌入在包括能够实现上述方法的所有特征并当被加载在计算机系统中时能实现这些方法的计算机程序产品中。本文中的“计算机程序”表示一组指令的任何语言、代码或符号的任何表达，其中，该表达被意图使具有信息处理能力的系统直接地执行特定功能或在以下处理中的一项或两者之后执行特定功能：转换为另一语言、代码或符号；以不同材料形式再现。

[0030] 虽然已经参照某些实施例描述了本公开，但是本领域的技术人员将要理解的是，在不脱离本公开的范围的情况下，可进行各种改变，并且可用等同物替代。此外，在不脱离本公开的范围的情况下，可进行一些修改以使特定的情况或材料适应本公开的教导。因此，不意图使本公开受限于公开的特定的实施例，而是意图本公开将包括落入所附权利要求范围内的所有实施例。

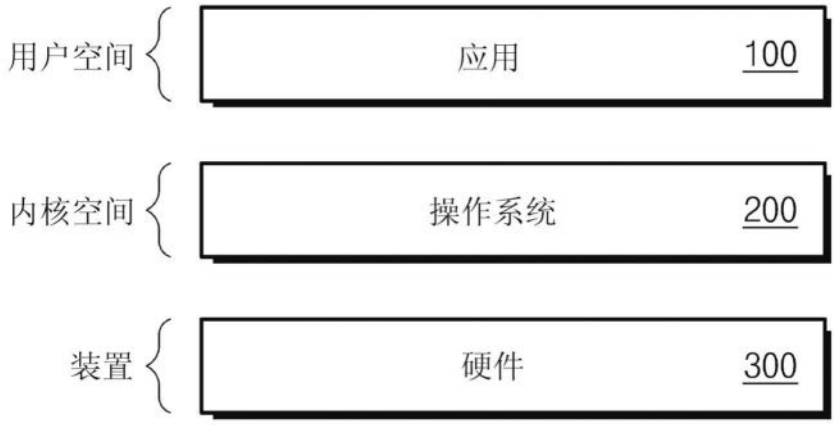


图1

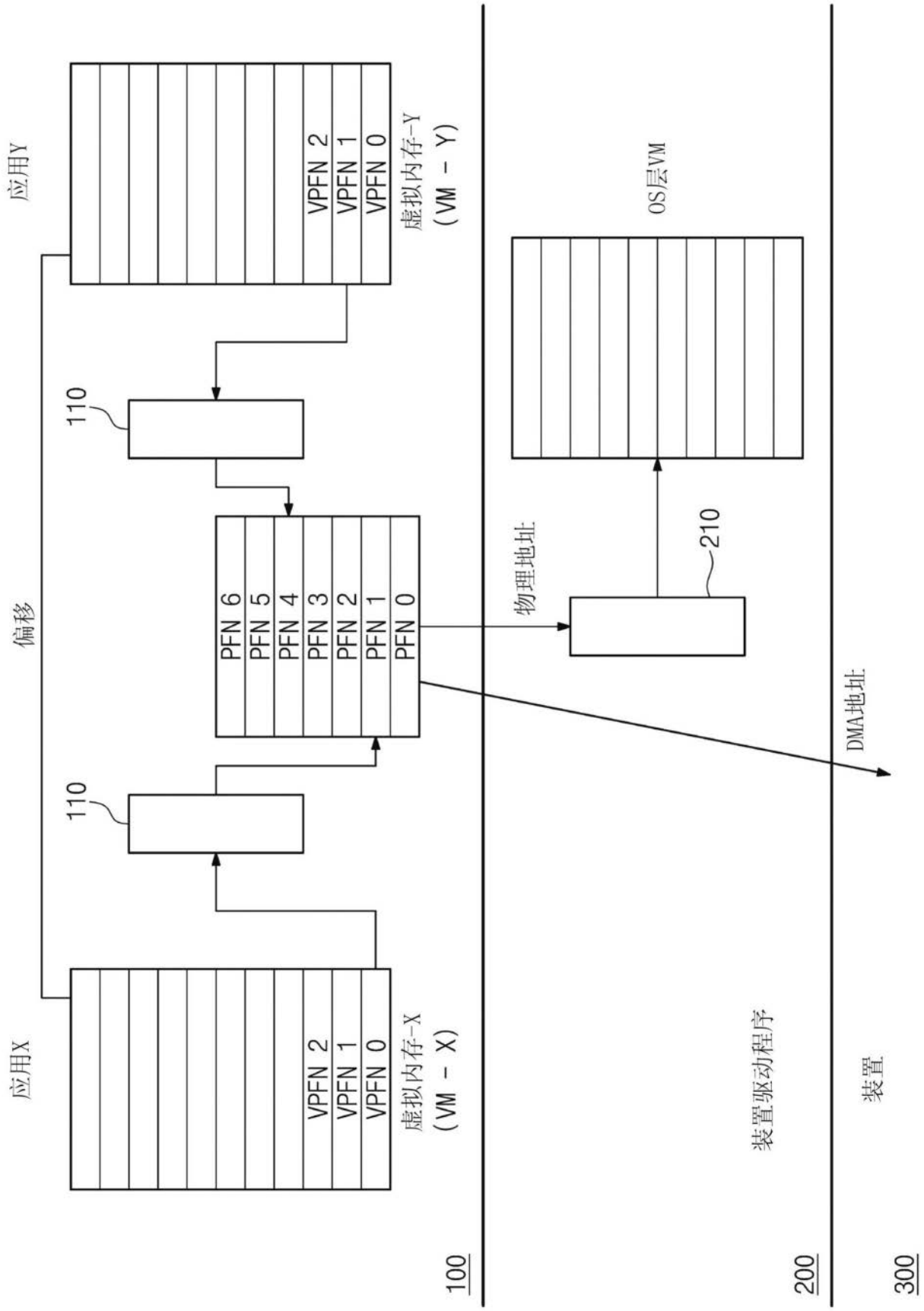

应用Y
虚拟内存-Y
(VM - Y)
VPFN 2
VPFN 1
VPFN 0
110
偏移
PFN 6
PFN 5
PFN 4
PFN 3
PFN 2
PFN 1
PFN 0
物理地址
OS层VM
210
110
DMA地址
应用X
虚拟内存-X
(VM - X)
VPFN 2
VPFN 1
VPFN 0
装置驱动程序
装置
100
200
300

图2A

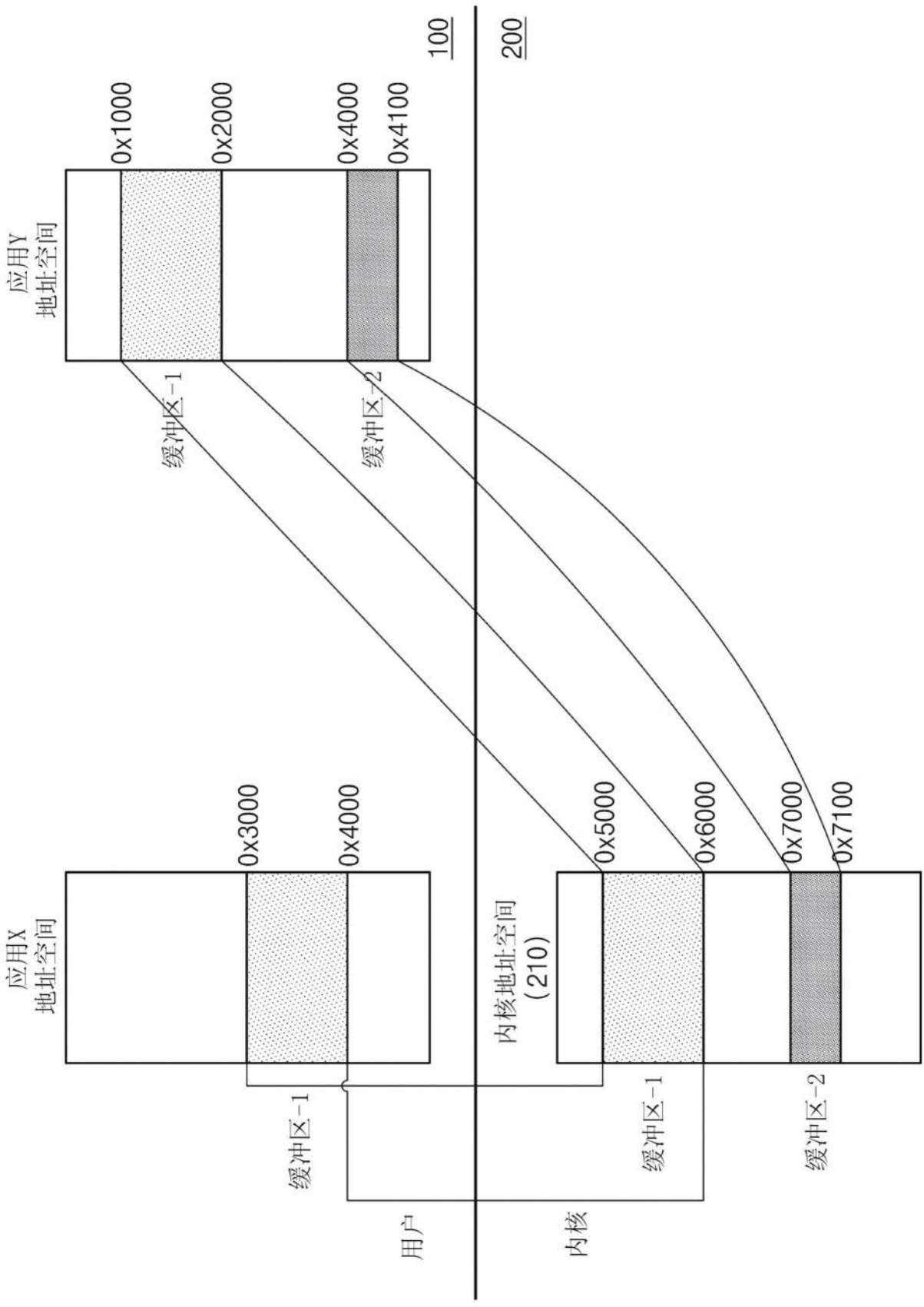

100
200
0x1000
0x2000
0x4000
0x4100
应用Y
地址空间
缓冲区-1
缓冲区-2
0x3000
0x4000
应用X
地址空间
0x5000
0x6000
0x7000
0x7100
内核地址空间
(210)
缓冲区-1
缓冲区-2
用户
内核

图2B

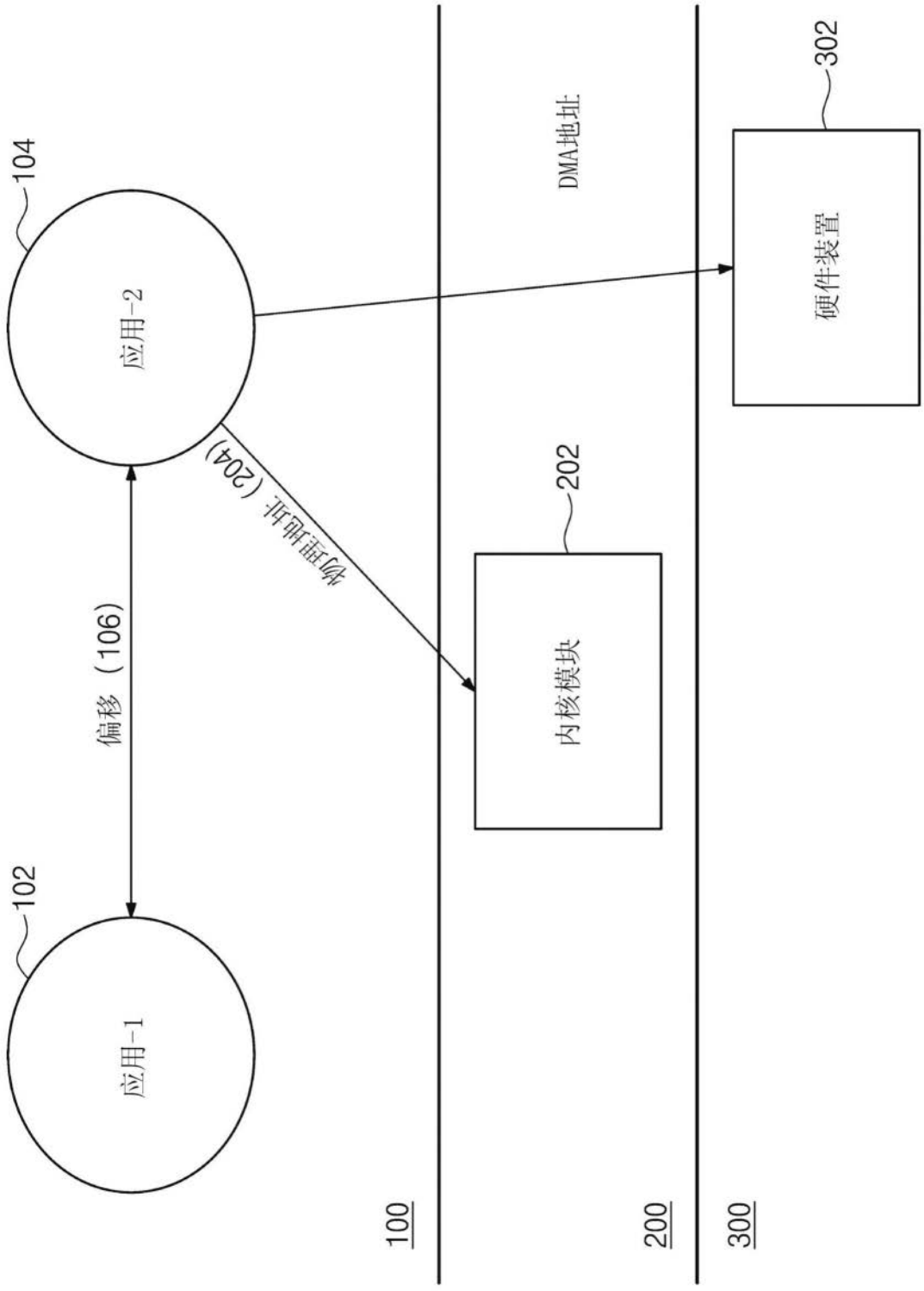

104
应用-2
102
应用-1
偏移（106）
物理地址（204）
DMA地址
202
内核模块
302
硬件装置
100
200
300

图3