地址转换和快速映射表构建方法、设备、存储介质及程序产品

本公开要求申请号为 202410330180.3 的中国专利申请的优先权，该中国专利申请于2024 年 03 月 20 日提交中国专利局，申请名称为“地址转换和快速映射表构建方法、设备、存储介质及程序产品”，其全部内容通过引用结合在本公开中。

技术领域

本公开涉及虚拟化技术领域，尤其涉及一种地址转换和快速映射表构建方法、设备、存储介质及程序产品。

背景技术

虚拟化技术是云计算的核心技术，它通过在宿主机的物理资源上创建一个或多个虚拟化实例（或称为客户机），实现对物理资源进行分割和集中管理的目的。宿主机具有宿主机操作系统（Host Operating System，Host OS）和宿主机看到的物理内存（实际物理内存），该物理内存的地址为 HPA（Host Physical Address，宿主机物理地址）。客户机具有客户机操作系统（Guest OS）和客户机看到的物理内存，该物理内存的地址称为 GPA（Guest Physical Address，客户机物理地址）。

在虚拟化环境下，通过 EPT（Extended Page Tables，扩展页表）进行内存管理，通过为每个客户机提供独立的多级页表结构，负责将客户机的 GPA 转换为宿主机的 HPA，同时实现不同客户机之间的隔离。通过 EPT 可以将客户机的 GPA 转换为宿主机的 HPA，并通过硬件支持的地址转换机制将访问引导到正确的物理内存位置，但是这种转换过程往往需要经过多级页表的查询来实现，存在效率低的问题。

发明内容

本公开的多个方面提供一种地址转换和快速映射表构建方法、设备、存储介质及程序产品，用以地址转换效率，实现快速地址转换。

本公开实施例提供一种地址转换方法，包括：预先构建与内存页表对应的第一快速映射表，内存页表包括至少一个客户机物理地址各自对应的多级页表；第一快速映射表中存储有至少一个客户机物理地址各自对应的最后一级页表项的地址信息；响应第一地址转换触发事件，确定需要进行地址转换的第一客户机物理地址，在第一快速映射表中查询第一客户机物理地址对应的最后一级页表项的地址信息；若查询到第一客户机物理地址对应的最后一级页表项的地址信息，根据第一客户机物理地址对应的最后一级页表项的地址信息和第一客户机物理地址的地址偏移量，生成第一客户机物理地址对应的宿主机物理地址。

本公开实施例还提供一种快速映射表构建方法，包括：将内存页表对应的客户机物理地址的地址空间划分为至少一个地址空间，针对至少一个地址空间分别创建基础结构体，基础结构体对应设定页表级别，内存页表包括至少一个客户机物理地址各自对应的多级页表；针对内存页表中任一客户机物理地址，根据任一客户机物理地址的地址信息和各基础结构体对应的地址空间，从各基础结构体中确定任一客户机物理地址对应的第一结构体；根据任一客户机物理地址对应的最后一级页表的级别与设定页表级别的关系，通过第一结构体记录任一客户机物理地址对应的最后一级页表项的地址信息，以构建与内存页表对应的第一快速映射表，第一快速映射表中存储有至少一个客户机物理地址各自对应的最后一级页表项的地址信息。

本公开实施例还提供一种物理机，包括：存储器和处理器；存储器，用于存储计算机程序；处理器，与存储器耦合，用于执行计算机程序，以实现本公开实施例提供的地址转换和快速映射表构建方法中的各步骤。

本公开实施例还提供一种存储有计算机程序的计算机可读存储介质，当计算机程序被处理器执行时，致使处理器实现本公开实施例提供的地址转换和快速映射表构建方法中的各步骤。

本公开实施例还提供一种计算机程序产品，包括计算机程序/指令，当计算机程序/指令被处理器执行时，致使处理器实现本公开实施例提供的地址转换和快速映射表构建方法中的各步骤。

在本公开实施例中，针对内存页表构建适配的第一快速映射表，在第一快速映射表中存储内存页表中至少一个客户机物理地址各自对应的最后一级页表项的地址信息，这样，在针对客户机物理地址进行地址转换时，可以先行查询内存页表对应的第一快速映射表，若在第一快速映射表中查询到该客户机物理地址对应的最后一级页表项的地址信息，则可以直接根据客户机物理地址对应的最后一级页表项的地址信息和客户机物理地址的地址偏移量，生成客户机物理地址对应的宿主机物理地址，无需在内存页表包含的与该客户机物理地址对应的多级页表中逐级的进行查询，可提高地址转换效率，实现快速地址转换。

在本公开一些可选实施例中，还针对内存页表构建适配的第二快速映射表，在第二快速映射表中存储至少一个宿主机物理地址到对应客户机物理地址的地址转换信息，用以支持宿主机物理地址到对应客户机物理地址的地址转换，解决基于内存页表只能实现客户机物理地址到宿主机物理地址的地址转换的不足，满足各种虚拟化环境中从宿主机物理地址到客户机物理地址之间的地址转换需求。

附图说明

此处所说明的附图用来提供对本公开的进一步理解，构成本公开的一部分，本公开的示意性实施例及其说明用于解释本公开，并不构成对本公开的不当限定。在附图

中：

图 1a 为本公开示例性实施例提供的地址转换方法的流程示意图；

图 1b 为本公开一示例性实施例提供的快速映射表构建方法的流程示意图；

图 2a 为本公开示例性实施例提供的一种物理机的结构示意图；

图 2b 为本公开示例性实施例提供的另一种物理机的结构示意图；

图 2c 为本公开示例性实施例提供的图 1b 所示物理机工作在根模式下的系统结构图；

图 2d 为本公开示例性实施例提供的图 1b 所示物理机工作在非根模式下的系统结构图；

图 3a 为本公开示例性实施例提供一种将宿主机操作系统从根模式切换到非根模式的流程示意图；

图 3b 为本公开示例性实施例提供一种创建内存页表的流程示意图；

图 3c 为本公开示例性实施例提供一种创建信息承载对象的流程示意图；

图 3d 为本公开示例性实施例提供一种模式切换的流程示意图；

图 4 为本公开示例性实施例提供地址转换装置的结构示意图；

图 5 为本公开一示例性实施例提供的快速映射表构建装置的结构示意图；

图 6 为本公开一示例性实施例提供的物理机的结构示意图。

具体实施方式

为使本公开的目的、技术方案和优点更加清楚，下面将结合本公开具体实施例及相应的附图对本公开技术方案进行清楚、完整地描述。显然，所描述的实施例仅是本公开一部分实施例，而不是全部的实施例。基于本公开中的实施例，本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例，都属于本公开保护的范围。

需要说明的是，本公开所涉及的用户信息（包括但不限于用户设备信息、用户个人信息等）和数据（包括但不限于用于分析的数据、存储的数据、展示的数据等），均为经用户授权或者经过各方充分授权的信息和数据，并且相关数据的收集、使用和处理需要遵守相关地区的相关法律法规和标准，并提供有相应的操作入口，供用户选择授权或者拒绝。

在虚拟化环境中，通过内存页表进行 GPA 到 HPA 之间的地址转换，可选地，内存页表可以实现为 EPT，在本公开下述实施例中，以内存页表实现为 EPT 为例进行描述，但不限于此。其中，EPT 中存储有至少一个 GPA 各自对应的多级页表。但是存在地址转换效率低的问题。针对该技术问题，在本公开实施例中，针对 EPT 构建第一快速映射（Rapid-Mapping，RMAP）表，第一 RMAP 表中存储有至少一个 GPA 各自对应的最后一级页表项的地址信息，用以建立 GPA 与 HPA 之间的快速映射关系，提高

地址转换效率，实现地址快速转换。

进一步，EPT 仅支持从 GPA 到 HPA 的地址转换，无法支持 HPA 到 GPA 的反向地址转换，在本公开实施例中，针对 EPT 创建第二 RMAP 表，第二 RMAP 表中存储有至少一个 HPA 到对应 GPA 的地址转换信息，用以建立 HPA 到 GPA 之间的快速映射关系，提高地址转换效率，实现地址快速转换。

进一步，在本公开实施例中，还详细介绍了第一 RMAP 表和第二 RMAP 表的构建过程、管理过程以及与 EPT 的状态同步更新过程，具体可参见后续实施例中的详细描述。

在此说明，本公开实施例提供的技术方案应用于虚拟化环境中，虚拟化环境是指采用虚拟化技术对宿主机上的物理资源进行虚拟化处理并在处理后的虚拟化资源上创建一个或多个虚拟化实例所产生的运行环境。在本公开实施例中，相对于宿主机，将承载于虚拟化资源上的虚拟化实例称为客户机。客户机具有自己的 OS 和 GPA，宿主机具有自己的 OS 和 HPA。

在本公开实施例中，并不限定虚拟化环境的实现方式。在一可选实施例中，虚拟化环境的实现方式 A 为：在宿主机 OS 中增设虚拟化管理器（hypvisor），由虚拟化管理器对宿主机上的硬件资源进行虚拟化处理，并在处理后的虚拟化资源上创建一个或多个虚拟机。在该虚拟化环境中，客户机的 GPA 与宿主机的 HPA 之间是通过 EPT 中至少一个 GPA 各自对应的多级页表进行地址转换。

在另一可选实施例中，虚拟化环境的实现方式 B 为：在宿主机的硬件资源和宿主机操作系统之间增设虚拟化层，至少在虚拟化层中实现虚拟资源提供功能，虚拟化层将运行中的宿主机操作系统从根模式（root）切换到非根模式（non-root），在非根模式下提供至少一种虚拟资源并在虚拟资源上至少运行宿主机操作系统在根模式下因资源不足而无法运行的应用，在无需重新安装操作系统的情况下，可以针对运行态的宿主机操作系统灵活实现虚拟资源提供功能，解决由于未开启或缺失系统功能等原因导致应用程序因资源不足而无法运行的问题，支撑更多应用的运行，有利于扩展宿主机操作系统的能力。

其中，根模式是宿主机操作系统对硬件资源具有直接访问权限的工作模式，非根模式是由虚拟化层对硬件资源进行虚拟化并代替宿主机操作系统对虚拟化后的资源进行调度和访问的工作模式。在非根模式下虚拟化层对硬件资源的访问是受限的。相应地，宿主机操作系统在非根模式下也可以被称为客户机操作系统。虚拟化层维护有 EPT，内存页表用于存储非根模式下的客户机物理地址与根模式下的宿主机物理地址之间的映射关系。虚拟化层可以基于 EPT 的多级页表，将 GPA 转换成 HPA。

在本公开实施例中，无论是在方式 A 实现的虚拟化环境中，还是在方式 B 实现的虚拟化环境中，都可以针对 EPT 创建第一 RMAP 表，基于第一 RMAP 表实现从 GPA 到 HPA 的快速映射，进一步，还可以创建第二 RMAP 表，基于第二 RMAP 表实现从

HPA 到 GPA 的反向映射，解决 EPT 无法实现从 HPA 到 GPA 的反向映射的问题，满足从 HPA 到 GPA 的反向映射问题。

在方式 A 中，创建第一 RMAP 表、基于第一 RMAP 表进行地址转换的过程以及创建第二 RMAP 表和基于第二 RMAP 表进行地址转换的过程可由宿主机 OS 完成，具体可由宿主机 OS 中的虚拟化管理器；在方式 B 中，创建第一 RMAP 表、基于第一 RMAP 表进行地址转换的过程以及创建第二 RMAP 表和基于第二 RMAP 表进行地址转换的过程可由虚拟化层在非根模式下完成。无论是方式 A 中的宿主机 OS，还是方式 B 中的虚拟化层，其创建第一 RMAP 表、基于第一 RMAP 表进行地址转换的过程以及创建第二 RMAP 表和基于第二 RMAP 表进行地址转换的过程相同或相似。

以下结合附图，对本公开实施例提供的一种解决方案进行详细说明。

图 1a 为本公开示例性实施例提供的一种地址转换方法的流程示意图。如图 1a 所示，该方法包括：

R11、预先构建与 EPT 对应的第一 RMAP 表，EPT 包括至少一个 GPA 各自对应的多级页表；第一 RMAP 表中存储有至少一个 GPA 各自对应的最后一级页表项的地址信息；

R12、响应第一地址转换触发事件，确定需要进行地址转换的第一 GPA，在第一 RMAP 表中查询第一 GPA 对应的最后一级页表项的地址信息；

R13、若查询到第一 GPA 对应的最后一级页表项的地址信息，根据第一 GPA 对应的最后一级页表项的地址信息和第一 GPA 的地址偏移量，生成第一 GPA 对应的 HPA。

在本实施例中，EPT 一方面采用多级页表结构，另一方面 EPT 采用混合映射粒度，不仅支持小页粒度，还支持大页粒度，可以满足不同物理页映射的需求，提高内存映射的灵活度。具体地，一方面 EPT 包括至少一个 GPA 各自对应的多级页表，多级页表可以是 2 级内存页表、4 级内存页表或 5 级内存页表等。以 EPT 包括 4 级内存页表为例，EPT 包括：页全局目录（Page Global Directory，PGD）、页上级目录（Page Upper Directory，PUD）、页中间目录（Page Middle Directory，PMD）以及页表（Page Table Entry，PTE）。PGD 中的页表项 pgd_t 中记录的是下一级页表（如，PUD）的地址信息，PUD 中的页表项 pud_t 中记录的是下一级页表（如，PMD）的地址信息，PMD 中的页表项 pmd_t 中记录的是下一级页表（如，PTE）的地址信息，PTE 中的页表项 pte_t 中记录的是 HPA 对应的物理页的起始地址。当然，若 PGD、PUD 以及 PMD 是最后一级页表，那么 PGD、PUD 以及 PMD 分别对应的页表项中记录的是 HPA 对应的物理页的起始地址。需要说明的是，GPA 地址可以被分成几份，每一份作为一个索引，得到多个索引值和一个地址偏移量，该地址偏移量是指相对于对应物理页的偏移量。以 EPT 包括 4 级内存页表为例，GPA 可以被分成 5 份，包括：PGD 级别的索引值、PUD 级别的索引值、PMD 级别的索引值、PTE 级别的索引值以及地址偏移量。通常情况下，基于 GPA 地址中的各索引值和地址偏移量查询 EPT 可以得到该 GPA 对应的 HPA。

另一方面，针对不同的地址空间（GPA）以不同页面粒度创建 EPT。例如，一些

地址空间通过小页粒度创建EPT的多级页表，若小页粒度为4k（对应PTE页面粒度），则该地址空间对应的4级页表，即PGD、PUD、PMD和PTE，最后一级页表为PTE；另一些地址空间可以采用大页粒度创建EPT，若大页粒度为2m（对应PMD页面粒度），则该地址空间对应3级页表，即PGD、PUD和PMD，最后一级页表为PMD；又一些地址空间可以采用大页粒度创建EPT，若大页粒度为1G（对应PUD页面粒度），则该地址空间对应2级页表，即PGD和PUD，最后一级页表为PUD。也就是说，EPT采用混合映射粒度，EPT中任一GPA对应的最后一级页表可以是PGD、PUD、PMD或PTE。关于EPT页表的详细介绍，可参见后续实施例，在此不再赘述。

为了方便记录GPA和HPA之间的快速映射关系，在本实施例中，提供一种第一RMAP表的数据结构，该数据结构存储有至少一个GPA各自对应的最后一级页表项的地址信息，从而绕过EPT中的多级页表转换，提高地址转换效率。

在本实施例中，由于EPT页表中的各GPA的最后一级页表可能是混合粒度，第一RMAP表中存储的最后一级页表项的地址信息可能是PGD、PUD、PMD和PTE中的至少一种页表的地址信息。

在本实施例中，预先构建与EPT对应的第一RMAP表的时机不作限定。例如，可以是创建EPT完毕之后，根据GPA与EPT的最后一级页表之间的对应关系创建第一RMAP表。又例如，可以在创建EPT的最后一级页表时，调用第一RMAP表的添加函数，建立对应的第一RMAP表，这样第一RMAP表就会随着EPT的建立而建立。

在本实施例中，维护有地址转换触发事件，为了便于区分和描述，将触发GPA转换为HPA的地址转换触发事件称为第一地址转换触发事件。第一地址触发事件可以是针对GPA的内存映射管理、内存冷热管理事件以及内存交换管理事件等。

相应地，在第一地址转换触发事件时，可以响应第一地址转换触发事件，确定需要进行地址转换的第一GPA，在第一RMAP表中查询第一GPA对应的最后一级页表项的地址信息；若查询到第一GPA对应的最后一级页表项的地址信息，根据第一GPA对应的最后一级页表项的地址信息和第一GPA的地址偏移量，生成第一GPA对应的HPA。例如，对第一GPA对应的最后一级页表项的地址信息进行解析，可以得到最后一级页表项中记录宿主机物理地址的起始地址，宿主机物理地址的对齐值（HPA_align），第一GPA的地址偏移量是指相对于该起始地址的偏移量，因此，该HPA_ align加上第一GPA的地址偏移量，可以生成第一GPA对应的HPA。

在一可选实施例中，可以针对EPT对应的GPA地址空间创建基础结构体，基于基础结构体，预先构建与EPT对应的第一RMAP表，相较于通过EPT逐级进行地址转换，基于基础结构体进行地址转换无需经过多级页表，地址转换效率较高。

具体地，考虑到EPT存在多种不同页表级别，可以从中选择设定页表级别作为基础页表级别，设定页表级别可以是任一页表级别，例如，设定页表级别可以是PUD页表级别、PMD页表级别、PTE页表级别或者PGD页表级别等。将EPT对应的GPA

地址空间划分为至少一个地址空间，针对至少一个地址空间分别创建基础页表级别（即设定页表级别）对应的基础结构体。其中，可以按照设定页表级别对应的页面粒度对EPT对应的GPA地址空间进行划分。例如，设定页表级别为PMD页表级别，PMD页表的粒度是2m，可以将EPT对应的GPA地址空间以2m的粒度划分为至少一个地址空间，针对每个2m的地址空间创建基础结构体。又例如，设定页表级别为PUD页表级别，PUD页表的粒度是1G，可以将EPT对应的GPA地址空间以1G的粒度划分为至少一个地址空间，针对每个1G的地址空间创建基础结构体。

针对EPT中任一GPA，根据任一GPA的地址信息和各基础结构体对应的地址空间，从各基础结构体中确定任一GPA对应的第一结构体。其中，各基础结构体对应的地址空间可以理解为一个地址范围，该地址范围的大小与设定页表级别对应的页面粒度对应。而任一GPA是一个地址信息，该地址信息可以落入某一基础结构体对应的地址空间中，为了便于区分和描述，将该任一GPA的地址信息所属的地址空间对应的基础结构体称为第一结构体。

根据任一GPA对应的最后一级页表的级别与设定页表级别的关系，通过第一结构体记录任一GPA对应的最后一级页表项的地址信息。其中，最后一级页表的级别与设定页表级别可以相同，或者最后一级页表的级别是设定页表级别的上一级别或下一级别。另外，任一GPA对应的最后一级页表项的地址信息可以由第一结构体记录，或者任一GPA对应的最后一级页表项的地址信息可以由第一结构体的扩展的第二结构体记录。关于第二结构体的详细介绍可参见前述实施例。

可选地，根据任一GPA对应的最后一级页表的级别与设定页表级别的关系，通过第一结构体记录该任一GPA对应的最后一级页表项的地址信息的实施方式不作限定，本可选实施例提供的方法是：若最后一级页表的级别与设定页表级别相同，则采用第一结构体记录最后一级页表项的地址信息；若最后一级页表的级别与设定页表级别不相同，则通过扩展映射表示不同于设定页表级别的其它页表级别，例如，针对第一结构体，扩展出第二结构体，在第二结构体中记录最后一级页表项的地址信息，在第一结构体中记录第二结构体的地址信息。

下面进行示例性说明。

示例C1：若任一GPA对应的最后一级页表的级别与设定页表级别相同，则认为无需对第一结构体进行扩展，直接采用第一结构体记录该最后一级页表项的地址信息，将任一GPA对应的最后一级页表项的地址信息添加到第一结构体中的信息字段中，并设置第一结构体中的标志信息为第一值，第一值表示第一结构体的信息字段中存储的是最后一级页表项的地址信息。其中，基础结构体可以是快速映射头部（rmap_head）结构体，例如，struct rmap_head {unsigned long val}，其中，struct是结构体，unsigned long表示数据类型为无符号长型，val为信息字段。

示例C2：若任一GPA对应的最后一级页表的级别与设定页表级别不相同，则认

为可以对第一结构体进行扩展，生成与任一 GPA 对应的最后一级页表的级别适配的第二结构体，将任一 GPA 对应的最后一级页表项的地址信息添加到第二结构体中的信息字段中，并将第二结构体的地址信息添加到第一结构体中的信息字段中，设置第一结构体中的标志信息为第二值，第二值表示第一结构体的信息字段中存储的是第二结构体的地址信息。

其中，需要说明的是，第一结构体中的标志信息主要用于标识第一结构体中的信息字段中存储的地址信息的类型。若该标志信息为第一值，表示第一结构体的信息字段中存储的是最后一级页表项的地址信息；若该标志信息为第二值，表示第一结构体的信息字段中存储的是第二结构体的地址信息。第一值和第二值的取值不同。例如，第一值和第二值可以通过不同信息字段中的不同比特位来表示，哪个比特位的值为 1 表示哪个值是有效的。又例如，可以通过信息字段中低三位的作为标志信息，低三位的取值范围是 0-7，第一值和第二值可以是 0-7 中的任意两个不同的值。

进一步可选地，创建与任一 GPA 对应的最后一级页表的级别适配的第二结构体的实施方式不作限定，下面进行示例性说明：将最后一级页表项的地址信息写入第二结构体中，将第二结构体的地址信息写入基础结构体中；这种情况下，pud 中的所有 GPA 对应的基础结构体可以共享同一第二结构体，无需针对每个基础结构体创建第二结构体

示例 D1： 在任一 GPA 对应的最后一级页表的级别为设定页表级别的上一级别的情况下，在一可选实施例中，可以针对最后一级页表创建多个基础结构体，针对每个基础结构体，生成最后一级页表的级别适配的与第二结构体，在每个第二结构体中记录最后一级页表项的地址信息，将每个第二结构体的地址信息写入相应基础结构体中。在另一可选实施例中，由于每个第二结构体中均记录有最后一级页表项的地址信息，因此，多个基础结构体可以共享同一第二结构体，在每个基础结构体中记录该共享的第二结构体的地址信息。

例如，设定页表级别为 PMD 级别，即基础结构体对应于 PMD 页表级别，任一 GPA 对应的最后一级页表的级别为设定页表级别的上一级别，即 PUD 级别，一个 PUD 大页下包括 512 个 PMD 页面，每个 PMD 页面对应一个基础结构体，一个 PUD 大页对应 512 个基础结构体，多个基础结构体可以共享同一第二结构体，第二结构体中存储有该 PUD 页面的地址信息，而每个基础结构体中的信息字段中存储的均是该共享第二结构体的地址信息。

基于上述，在任一 GPA 对应的最后一级页表的级别为设定页表级别的上一级别的情况下，可以判断是否已经存在第二结构体；若不存在第二结构体，创建与任一 GPA 对应的最后一级页表的级别适配的新结构体，作为第二结构体，将第二结构体的地址信息添加到该任一 GPA 对应的第一结构体中的信息字段中，设置该第一结构体中的标志信息为第二值。

可选地，若已存在第二结构体，则将已存在的第二结构体的地址信息添加到与任一 GPA 对应的第一结构体中的信息字段中，设置该第一结构体中的标志信息为第二值。

其中，在任一 GPA 对应的最后一级页表的级别为设定页表级别的上一级别的情况下，第二结构体可以表示为 rmap_cmpt {unsigned long val; atomic_t num; }。struct 是结构体，unsigned long 表示数据类型为无符号长型，val 为信息字段，atomic_t num 表示 GPA-HPA 快速映射的数量。

示例 D2：在任一 GPA 对应的最后一级页表的级别为预设页表级别的下一级别的情况下，创建与任一 GPA 对应的最后一级页表的级别适配的新结构体，作为第二结构体。

例如，预设页表级别为 PMD 页表级别，任一 GPA 对应的最后一级页表的级别为预设页表级别的下一级别，即 PTE 页表级别，该预设页表级别的页表可以扩展出多个下一级别的页表，如，PDM 级别的 2m 页表可以映射出 512 个 PTE 级别的页表。相应地，该任一 GPA 对应的第一结构体可以扩展多个（如，512 个）第二结构体，多个第二结构体用于存放预设页表级别的页表（如，PMD 页表）映射出的下一级别页表（PTE 页表）的地址信息。进一步，可以采用统一结构体对多个第二结构体进行组织管理，统一结构体可以采用数组的方式对多个第二结构体进行组织管理，将每个第二结构体中存储的地址信息看作是该数组中的一个元素。其中，统一结构体可以是后续快速映射扩展（rmap_extd）结构体，第二结构体中存储的地址信息是该 rmap_extd 结构体中包含数组的元素其中，统一结构体表示为 rmap_extd{unsigned long val[VAL_EXTD_NR]; atomic_t num; }。其中，struct 是结构体，unsigned long 表示数据类型为无符号长型，atomic_t num 表示 GPA-HPA 快速映射的数量。val[VAL_EXTD_NR]为数组，第二结构体中存储的地址信息是该 val[VAL_EXTD_NR]数组的元素。

其中，可以确定下一级别页表的粒度（如，4k）在预设页表级别的页表（如，2m）中的偏移量，创建该偏移量对应的新的结构体，将该新的结构体作为该任一 GPA 对应的第二结构体。例如，该任一 GPA 位于 2m 中第 3 个 4k 页，则创建该偏移量 3 对应的新的结构体，将该新的结构体作为该任一 GPA 对应的第二结构体。还可以将任一 GPA 对应的最后一级页表（如，PTE 页表）的地址信息添加到该第二结构体的信息字段中，并设置第二结构体中的标志信息为第二值。

在一可选实施例中，在已经存在第一 RMAP 表的情况下，可以针对任一 GPA 进行查询操作、增加操作、删除操作、修改操作、扩展操作以及收缩操作等操作。下面进行示例性说明。

示例 E1：以针对第一 GPA 执行查询操作为例进行描述。

其中，查询第一 GPA 对应 HPA 的情况不作限定，例如，针对第一 GPA 读写数据时，可以触发查询第一 GPA 对应 HPA，从而基于第一 RMAP 表快速获取第一 GPA 对应的 HPA。

在查询第一 GPA 对应 HPA 的情况下，在第一 RMAP 表中查询第一 GPA 对应的最后一级

页表项的地址信息；若查询到第一 GPA 对应的最后一级页表项的地址信息，根据第一 GPA 对应的最后一级页表项的地址信息和第一 GPA 的地址偏移量， 生成第一 GPA 对应的 HPA。可选地，若未查询到第一 GPA 对应的最后一级页表项的地址信息，则认为该第一 GPA 不存在 HPA，可以触发缺页异常。

在一可选实施例中，根据第一 GPA 的地址信息和第一 RMAP 表中包含的各基础结构体对应的地址空间，从各基础结构体中确定第一 GPA 对应的第一结构体，第一结构体是第一 RMAP 表中包含的各基础结构体中第一 GPA 对应的基础结构体；通过第一 GPA 对应的第一结构体，获取第一 GPA 对应的最后一级页表项的地址信息。根据第一GPA对应的最后一级页表项的地址信息和第一GPA的地址偏移量,生成第一GPA对应的 HPA。

可选地，通过第一 GPA 对应的第一结构体，获取第一 GPA 对应的最后一级页表项的地址信息，包括：从第一结构体中获取标志信息；根据标志信息的不同，获取第一 GPA 对应的最后一级页表项的地址信息的方式也有所不同。若标志信息为第一值，则表示第一结构体的信息字段中记录的是最后一级页表项的地址信息，可以直接从第一结构体中的信息字段中获取第一 GPA 对应的最后一级页表项的地址信息；若标志信息为第二值，则表示第一结构体的信息字段中记录的是第二结构体的地址信息，第二结构体中存储的是最后一级页表项的地址信息，可以从第一结构体中的信息字段中获取第二结构体的地址信息，根据第二结构体的地址信息，从第二结构体的信息字段中获取第一 GPA 对应的最后一级页表项的地址信息。

进一步可选地，第一结构体的信息字段中存储的是统一结构体的地址信息以及第二结构体在统一结构体数组中的偏移量（offset，oft），该偏移量 oft 表示该任一 GPA 地址在设定页表级别的页表中最后一级页面粒度的偏移量，例如，设定页表级别为 PMD 级别，最后一级页面粒度为 PTE 级别，则该偏移量 oft 是指该任一 GPA 地址在 PMD 级别的 2m 中 4k 粒度的偏移量，若偏移量为 oft3，表示该任一 GPA 位于 2m 中第 3 个 4k 地址空间中。相应地，从第二结构体的信息字段中获取第一 GPA 对应的最后一级页表项的地址信息，包括：根据该偏移量 oft，选取统一结构体（如，rmap_extd）数组中第 oft 个元素，该元素的取值即为第一 GPA 对应的最后一级页表项的地址信息。

示例 E2：以针对第二 GPA 执行增加操作为例进行说明。

其中，增加第二 GPA 的情况不作限定，例如，在 GPA 不足的情况下，可以增加 GPA，这时，可以触发针对 GPA 地址的增加操作，在本实施例中，可以添加一个或多个 GPA，下面以添加第二 GPA 为例进行描述。

在一可选实施例中，在 EPT 中新增第二 GPA 的情况下，在第一 RMAP 表中添加第二 GPA 对应的最后一级页表项的地址信息。

可选地，根据第二 GPA 的地址信息和第一 RMAP 表中包含的各基础结构体对应的地址空间，从各基础结构体中确定第二 GPA 对应的第一结构体；若确定未通过第一

结构体记录第二 GPA 的对应的最后一级页表项的地址信息，表示第一 RMAP 表中并未记录第二 GPA 的对应的最后一级页表项的地址信息，则根据第二 GPA 的对应的最后一级页表的级别与设定页表级别的关系，通过第一结构体记录第二 GPA 对应的最后一级页表项的地址信息。详细实施方式可参见前述“根据任一 GPA 对应的最后一级页表的级别与设定页表级别的关系，通过第一结构体记录任一 GPA 对应的最后一级页表项的地址信息”的介绍，在此不再赘述。

示例 E3：以针对第三 GPA 执行删除操作为例进行说明。

其中，删除 GPA 的情况不作限定，例如，需要对虚拟内存进行缩容的情况下，可以删除部分 GPA，并删除 GPA 与 HPA 之间的转换关系。下面以删除第三 GPA 为例进行描述。

在一可选实施例中，在 EPT 中删除第三 GPA 的情况下，在第一 RMAP 表中删除第三 GPA 对应的最后一级页表项的地址信息。

可选地，根据第三 GPA 的地址信息和第一 RMAP 表中包含的各基础结构体对应的地址空间，从各基础结构体中确定第三 GPA 对应的第一结构体；若第三 GPA 对应的最后一级页表的级别和第一结构体的标志信息指示的页表级别适配，则根据第三 GPA 对应的最后一级页表的级别与设定页表级别的关系，通过第一结构体删除第三 GPA 对应的最后一级页表项的地址信息。

其中，第三 GPA 对应的最后一级页表的级别和第一结构体的标志信息指示的页表级别适配可以理解为：最后一级页表的级别与标志信息指示的地址信息所属页表级别相同。其中，第一结构体的标志信息不但用于标识第一结构体的信息字段中存储的地址信息的类型，在该标志信息标识最后一级页表项的地址信息的情况下，该标志信息还可以标识第一结构体的信息字段中记录的地址信息所属的页表级别，即该地址信息是哪一页表级别的页表项的地址信息。例如，在该标志信息标识最后一级页表项的地址信息的情况下，标志信息为第二值，该第二值可以是 X0 或 X1，X1 表示该地址信息是设定页表级别的下一级别（如，PTE 级别）的页表项的地址信息，X0 表示该地址信息是设定页表级别的上一级别（如，PUD 级别）的页表项对应的地址信息。

示例 E4：以针对第四 GPA 执行修改操作为例进行说明。

其中，修改 GPA 对应的 HPA 地址的情况不作限定。例如，在执行内存交换操作的情况下，该 GPA 对应原 HPA 页面被交换到磁盘上，则可以将该 GPA 对应的原 HPA 修改为新的 HPA。下面以修改第四 GPA 为例进行介绍。

在一可选实施例中，在 EPT 中将第四 GPA 对应的最后一级页表的第一地址信息修改为第二地址信息的情况下，在第一 RMAP 表中将第四 GPA 对应的最后一级页表的第一地址信息修改为第二地址信息。

可选地，可以根据该第四 GPA 的地址信息和第一 RMAP 表中包含的各基础结构体对应的地址空间，从各基础结构体中确定第四 GPA 对应的第一结构体；通过第一结构体将第四

GPA 对应的最后一级页表项的第一地址信息修改为第二地址信息。

进一步可选地，从第一结构体中获取标志信息；若标志信息为第一值，则将第一结构体中的信息字段中记录的第一地址信息修改为第二地址信息；若标志信息为第二值，则从第一结构体中的信息字段中获取第二结构体的地址信息，根据第二结构体的地址信息，将第二结构体的信息字段中记录的第一地址信息修改为第二地址信息。

示例 E5：针对 EPT 中最后一级页表进行扩展，得到下一级页表。下面以针对最后一级页表进行扩展为例进行描述。

在一可选实施例中，在EPT中将最后一级页表扩展为下一级页表的情况下，在第一RMAP表中将最后一级页表中各GPA各自对应的最后一级页表项的地址信息修改为相应下一级页表的地址信息。

可选地，在 EPT 中清理最后一级页表中各页表项记录的地址信息（即 HPA 对应物理页的地址信息），以及清理第一 RMAP 表中最后一级页表中各 GPA 对应的最后一级页表项的地址信息；创建最后一级页表对应的下一级页表，在最后一级页表的页表项中记录下一级页表的地址信息，并在下一级页表的页表项中记录相应 HPA 对应的物理页的地址信息；在第一 RMAP 表中，根据下一级页表的级别与设定页表级别的关系，通过第一结构体记录最后一级页表中的各 GPA 对应的下一级页表项的地址信息。

示例 E6：针对 EPT 中最后一级页表进行收缩，得到上一级页表。下面以针对最后一级页表进行收缩为例进行描述。

在一可选实施例中，在 EPT 中地址连续的最后一级页表收缩为上一级页表的情况下，在第一 RMAP 表中清理最后一级页表项的地址信息，并添加最后一级页表中各 GPA 对应的上一级页表项的地址信息。

可选地，在 EPT 中清理地址连续的最后一级页表中各页表项记录的地址信息，以及清理第一 RMAP 表中最后一级页表中各 GPA 对应的最后一级页表项的地址信息；在 EPT 中创建最后一级页表对应的上一级页表，按照所述上一级页表项对应的页面粒度，在上一级页表的页表项中分别记录相应 HPA 对应的物理页的基地址；在第一 RMAP 表中，根据上一级页表的级别与设定页表级别的关系，通过第一结构体记录最后一级页表中的各 GPA 对应的上一级页表项的地址信息。

在一可选实施例中，考虑到 EPT 页表无法实现 HPA 到 GPA 的反向转换，而底层在进行内存管理时大多以 HPA 作为管理单元，以方便后续进行内存管理、冷热管理以及内存交换等。因此，需要一种 HPA 到 GPA 的快速映射关系。

在本可选实施例中，预先构建与 EPT 对应的第二 RMAP 表，第二 RMAP 表中存储有至少一个 HPA 到对应 GPA 的地址转换信息；该对应 GPA 的地址转换信息是指该对应 GPA 所属 GPA 页面的基地址，该 GPA 页面的大小与 HPA 页面的大小对应。例如，第二 RMAP 表中通过第三结构体存储 2m 粒度切分的 HPA 的地址信息，反向查找直接找到的 GPA 也是对应 2m 粒度切分的；至于其它粒度 4k 或 1G 需要下一级扩展结

构（如，第四结构体）存储。例如，若 EPT 页表中维护的最后一级页表项对应的 HPA 页面的大小为 4k,那么该 4k 大小的 HPA 页面对应于 4k 大小的 GPA 页面,在第二 RMAP 表中,通过第三结构体记录 2m 粒度的 GPA 页面,再进一步,通过对该 2m 粒度的 GPA 页面进行扩展，得到多个 4k 粒度的页面，通过第四结构体记录多个 4k 粒度的页面；若 EPT 页表中维护的最后一级页表项对应的 HPA 页面的大小为 2m，则该 2m 大小的 HPA 页面对应于 2m 大小的 GPA 页面，在第二 RMAP 表中，通过第三结构体记录该 2m 粒度的 GPA 页面；若 EPT 页表中维护的最后一级页表项对应的 HPA 页面的大小为 1G，则该 1G 大小的 HPA 页面对应于 1G 大小的 GPA 页面，将 1G 大小的 GPA 页面拆分为多个 2m 粒度的 GPA 页面，在第二 RMAP 表中，通过第三结构体记录每个 2m 粒度的 GPA 页面，通过第四结构体记录 2m 粒度的页面所属的 GPA 大页的地址信息，该 1G 页面下的 512 个 2m 粒度的页面共享同一 GPA 大页的地址信息。响应第二地址转换触发事件，确定需要进行地址转换的第一 HPA，在第二 RMAP 表中查询第一 HPA 到对应 GPA 的地址转换信息；根据第一 HPA 及其到对应 GPA 的地址转换信息和第一 HPA 的地址偏移量，得到第一 HPA 对应的 GPA。例如，通过该地址转换信息加上第一 HPA 的偏移量得到第一 HPA 对应的 GPA。其中，第二地址转换触发事件可以是针对 GPA 的内存映射管理、内存冷热管理事件以及内存交换管理事件等。

可选地，预先构建与 EPT 对应的第二 RMAP 表的实施方式，包括：将 EPT 对应的 HPA 地址空间划分为至少一个地址空间,针对至少一个地址空间分别创建基础结构体，基础结构体对应设定页面粒度，设定页面粒度可以是 4k、2m 或 1G 等。针对 EPT 中的任一 GPA 对应的 HPA，根据任一 HPA 的地址信息和各基础结构体对应的地址空间，从各基础结构体中确定任一 HPA 对应的第三结构体，第三结构体是第二 RMAP 表中与该任一 HPA 对应的基础结构体，可以认为该任一 HPA 位于第三结构体对应的地址空间中。例如，第三结构体可以是 rmap_head 结构体。根据任一 HPA 对应的页面粒度和设定页面粒度的关系，通过第三结构体记录任一 HPA 到对应 GPA 的地址转换信息。其中，任一 HPA 对应的页面粒度是已知的，该任一 HPA 对应的页面粒度可以与设定页面粒度相同，也可以与设定页面粒度不相同。

进一步可选地，若任一 HPA 对应的页面粒度与设定页面粒度相同，则通过第三结构体中的信息字段记录任一 HPA 到对应 GPA 的地址转换信息，并设置第三结构体中的第一标志信息为第三值，第三值表示第三结构体的信息字段中记录的是任一 HPA 到对应 GPA 的地址转换信息，第三值可以是 0 或 1 等。若任一 HPA 对应的页面粒度与设定页面粒度相同不相同，则创建与任一 HPA 对应的页面粒度适配的第四结构体，第四结构体对应的地址空间的粒度与任一 HPA 对应的页面粒度相同。例如，4k 的页面粒度对应的第四结构体可以是 rmap_extd，1G 的页面粒度对应的第四结构体可以是 rmap_cmpt。通过第四结构体中的信息字段记录任一 HPA 到对应 GPA 的地址转换信息，并将第四结构体的地址信息添加到第三结构体中的信息字段中，设置第三结构体中的

第一标志信息为第四值。其中，第四值表示第三结构体的信息字段中记录的是第四结构体的地址信息，第四值可以是 0 或 1。其中，第三值和第四值可以用第三结构体的信息字段中的低三位表示，例如，第三值和第四值可以是 0-7 中任意两个不同的值。或者第三值和第四值可以通过两个标志位来表示，标志位为 1，表示值有效。

进一步可选地，创建与任一 HPA 的页面粒度适配的第四结构体的实施方式，包括：在任一 HPA 的页面粒度为设定页面粒度的上一级的情况下，判断是否已经存在第四结构体；若不存在，创建与任一 HPA 的页面粒度适配的新结构体，作为第四结构体。在任一 HPA 的页面粒度为设定页面粒度定页表级别的下一级别的情况下，创建与任一 GPA 对应的最后一级页表的级别适配的新结构体，作为第四结构体。该实施例的详细介绍可参见创建与任一 GPA 的最后一级页表的级别适配的第二结构体的实施方式，再次不再赘述。例如，页面粒度按照级别由上到下一次可以是：512G、1G、2m 以及 4k。若任一 HPA 的页面粒度为 1G，设定页面粒度为 2m，则认为任一 HPA 的页面粒度为设定页面粒度的上一级。若任一 HPA 的页面粒度为 4k，设定页面粒度为 2m，则认为任一 HPA 的页面粒度为设定页面粒度的下一级。

在一可选实施例中，HPA 和 GPA 之间的映射可以是一一对应关系，也可以是一对多关系，例如，一个 HPA 对应有多个 GPA。相应地，通过第三结构体中的信息字段记录任一 HPA 到对应 GPA 的地址转换信息的实施方式，包括：在任一 HPA 对应多个 GPA 的情况下，创建第五结构体，该第五结构体是第三结构体的扩展，是在任一 HPA 的页面粒度与设定页面粒度相同的情况下，记录任一 HPA 到多个 GPA 的地址转换信息的结构体。将第五结构体的地址信息添加到第三结构体中的信息字段中，并将任一 HPA 到多个 GPA 的地址转换信息分别添加到第五结构体中，并在第三结构体中增设表示一对多关系的标志位；在任一 HPA 对应一个 GPA 的情况下，直接将任一 HPA 到对应 GPA 的地址转换信息添加到第三结构体中的信息字段中，无需设置表示一对多关系的标志位，默认 HPA 与 GPA 是一对一关系。

相应地，通过第四结构体中的信息字段记录任一 HPA 到对应 GPA 的地址转换信息，包括：在任一 HPA 对应多个 GPA 的情况下，则创建第六结构体，第六结构体是是第四结构体的扩展，是在任一 HPA 的页面粒度与设定页面粒度不相同的情况下，记录任一 HPA 到多个 GPA 的地址转换信息的结构体。将第六结构体的地址信息添加到第四结构体中的信息字段中，并将任一 HPA 到多个 GPA 的地址转换信息分别添加到第六结构体中，在第四结构体中增设表示为一对多关系的标志位；在任一 HPA 对应一个 GPA 的情况下，直接将任一 HPA 到对应 GPA 的地址转换信息添加到第四结构体中的信息字段中。

需要说明的是，第五结构体和第六结构体可以实现为 rmap_desc 结构体，即 rmap_desc {unsigned long val[VAL_DESC_NR]; struct rmap_desc *more; }。其中，val[VAL_DESC_NR]表示用于存储任一 HPA 到对应多个 GPA 的地址转换信息，该数

组中的每个元素表示任一 HPA 到对应一个 GPA 的地址转换信息，*more 指针可以指向下一个 rmap_desc，下一 rmap_desc 结构体中也包括 val[VAL_DESC_NR]数组，方便存储更多的 GPA 的地址转换信息。

下面以如下所示进行示例对基于第一 RMAP 表和第二 RMAP 表进行增加、查询、删除、修改、扩展以及收缩和扩展操作进行示例性说明，但并不限于此。

其中，EPT 页表是四级页表结构，包括 PGD、PUD、PMD 以及 PTE 四种页表，EPT 最后一级页表可能是 PGD、PUD、PMD 和 PTE 中的至少一种。相应地，最后一级页表项可以是 pgd_t、pud_t、pmd_t 或者 pte_t。基础结构体用 rmap_head 标识。PTE 级别的统一结构体为 rmap_extd，第二结构体中存储的最后一级页表项的地址信息是统一结构体中的 val[VAL_EXTD_NR]数组中的元素。PUD 级别的第二结构体为 rmap_cmpt。

一、针对第一 RMAP 表查询第一 GPA 对应的 HPA。

1、首先根据第一 GPA 的地址信息和第一 RMAP 表中包含的各基础结构体对应的地址空间，从各基础结构体中确定第一 GPA 对应的第一结构体（rmap_head）。其中，可以将第一 RMAP 表看作一个大数组，大数组中的元素为基础结构体，每个基础结构体对应的地址空间为 2m，因此，根据第一 GPA 的地址信息，以 2m 为粒度从大数组中选取第一 GPA 对应的基础结构体，将该选取的基础结构体作为第一结构体。

2、根据第一结构体 rmap_head 中信息字段（val）的标志信息选择不同的查询方式。

1）若标志信息为 rmap_cmpt 对应的第二值，表示第二 GPA 的最后一级页表项对应的页面粒度为 1G。

a)从 rmap_head 的信息字段(val)中解析出 rmap_cmpt 结构，即获取 rmap_cmpt 的地址信息；

b）将 rmap_cmpt（第二结构体）中的信息字段解析为新的 rmap_head，这里，rmap_head 是一个通用结构，新的 rmap_head 可以理解为 rmap_cmpt 的信息字段中包含的值；

c）判断新的 rmap_head 中的 val 值是否是页表项的地址信息。例如，可以根据 val 值中的标志信息，确定 val 值是否是页表项的地址信息。若是，则根据新的 rmap_head 中记录的第一 GPA 的最后一级地址信息，读取该地址信息中记录的物理页的基地址，将该基地址加上第一 GPA 的地址偏移量，生成第一 GPA 对应的 HPA，该地址偏移量是在 1G 内存页中的偏移量。若否，表示新的 rmap_head 中的 val 值具有下一级扩展结构，val 值为下一级扩展结构的地址信息，下一级扩展结构可以是一对多关系对应的结构。

2）若标志信息为 rmap_extd 对应的第二值，表示该第二 GPA 对应的最后一级页表项对应的页面粒度为 4k。

a）从 rmap_head 的信息字段中解析出 rmap_extd 结构，即获取 rmap_extd 的地址信息；

b）根据第二 GPA 在 2m 中 4k 粒度的偏移量 oft，从 val[VAL_EXTD_NR]数组中选取 rmap_extd 中第 oft 个 val 元素，即第二结构体的地址信息；

c）将该第 oft 个 val 元素解析为新的 rmap_head；

d）判断新的 rmap_head 中的 val 值是否是页表项的地址信息。若是，则根据新的 rmap_head 中记录的第一 GPA 的最后一级地址信息，读取该地址信息中记录的物理页的基地址，将该基地址加上第一 GPA 的地址偏移量，生成第一 GPA 对应的 HPA，该地址偏移量是在 4k 内存页中的偏移量。若否，表示新的 rmap_head 中的 val 值具有下一级扩展结构，val 值为下一级扩展结构的地址信息，下一级扩展结构可以是一对多关系对应的结构。

3）、若标志信息为 rmap_head 对应的第一值或者未设置标记信息，则最后一级页表项对应的页面粒度为 2m，不需要解析出下层结构。

a）判断新的 rmap_head 中的 val 值是否是页表项的地址信息。若是，则新的 rmap_head 中记录的第一 GPA 的最后一级地址信息，读取该地址信息中记录的物理页的基地址，将该基地址加上第一 GPA 的地址偏移量，生成第一 GPA 对应的 HPA，该地址偏移量是在 2m 内存页中的偏移量。若否，表示新的 rmap_head 中的 val 值具有下一级扩展结构，val 值为下一级扩展结构的地址信息，下一级扩展结构可以是一对多关系对应的结构。

本实施例中还包括针对第二 RMAP 表查询第一 HPA 对应的 GPA，该过程与第一 RMAP 表查询第一 GPA 对应的 HPA 的实施方式类似，在此不再赘述。

另外，由于存储 HPA 与 GPA 一对多的情况下，在第二 RMAP 表中查询第一 HPA 对应的 GPA 的情况下，还可以包括一对多映射的查找操作，下面重点介绍一对多相对于一对一不同的部分，其它不再赘述。

例如，针对第二 HPA 查找对应 GPA。若第三结构体中的标志信息为第四值，第四结构体为 rmap_cmpt，第四结构体 rmap_cmpt 中的标志信息指示存在一对多映射关系，第四结构体中存储的是第五结构体 rmap_desc 的地址信息，需要循环 rmap_desc 中记录的各个 val，判断 val 是否是页表项的地址信息，若是，则直接返回该 val 值。若否，则 val 值是下一级扩展结构的地址信息，可以从下一级扩展结构中获取 val。

其中，在查询 rmap_desc 时，可以获取 rmap_desc 的 val[VAL_DESC_NR]数组中的全部元素，根据每个元素记录的 GPA 的地址转换信息以及第一 HPA 的偏移量计算对应的 GPA。当然，也可以按照设定的筛选条件，返回数组中的某一个元素，将该元素记录的 GPA 的地址转换信息加上第一 HPA 的偏移量，得到该第一 HPA 对应的 GPA。其中，若 val[VAL_DESC_NR]的长度为 5，表示该数组可以存储 5 个元素，该数组通过*more 指针连接有后续数组，若要获取数组的第 13 个元素，则可以通过第一个

rmap_desc 中的 more 指针，获取第二个 rmap_desc 的地址信息，基于该第二个 rmap_desc 的地址信息，确定第二个 rmap_desc 中的 more 指针指向的第三个 rmap_desc 的地址信息，根据该第三个 rmap_desc 的地址信息。获取第三个 rmap_desc 的 val[VAL_DESC_NR] 中的第 3 个元素，该第 3 个元素即是第一 HPA 对应的 GPA 的地址转换信息，再加上第一 HPA 的地址偏移量，可以得到第一 HPA 对应的 GPA。若 val[VAL_DESC_NR]的长度为 5，要获取数组的第 3 个元素，则可以直接从第一个 rmap_desc 的 val[VAL_DESC_NR]中获取第 3 个元素，即第一 HPA 对应的 GPA 的地址转换信息，该地址转换信息加上第一 HPA 的地址偏移量，可以得到第一 HPA 对应的 GPA。

二、针对第一 RMAP 表增加第二 GPA 对应的 HPA。

1、首先根据第二 GPA 的地址信息和第一 RMAP 表中包含的各基础结构体对应的地址空间，从各基础结构体中确定第二 GPA 对应的第一结构体（rmap_head）。其中，可以将第一 RMAP 表看作一个大数组，大数组中的元素为基础结构体，每个基础结构体对应的地址空间为 2m，因此，根据第二 GPA 的地址信息，以 2m 为粒度从大数组中选取第二 GPA 对应的基础结构体，将该选取的基础结构体作为第一结构体。

2、确定第二 GPA 的最后一级页表的页表级别（level）；获取第一结构体的标志信息，判断标志信息指示第一结构体的信息字段中存储的地址信息是否对应于页表级别（level）的结构体，若是，则认为已经存在页表级别（level）的结构体，若否，则认为不存在页表级别（level）对应的结构体，则创建页表级别（level）对应的结构体。例如，页表级别 level 为 PUD 级别，但是第一结构体的标志信息未设置为 rmap_cmpt 对应的值，则分配一个 rmap_cmpt 结构体，并 rmap_cmpt 结构体写入第一结构体的信息字段中。

3、判断页表级别 level 对应的结构体中是否存在该第二 GPA 对应的最后一级页表项的地址信息。若存在，则直接返回该地址信息；

4、若不存在，为该第二 GPA 增加 HPA。此时，可以根据第一结构体 head rmap 中信息字段（val）的标志信息选择不同的增加方式。

1）若标志信息为 rmap_cmpt 对应的第二值，表示第二 GPA 的最后一级页表项对应的页面粒度为 1G。

需要说明的是，首次针对 GPA 建立 1G 粒度的映射时，可以申请一个 rmap_cmpt 结构，在该 rmap_cmpt 结构中记录该 1G 的 GPA 对齐的 HPA 的地址信息，并增加计数；将 1G 内存页包含的 512 个 rmap_head 的地址信息与该 rmap_cmpt 建立关联，即，在每个 rmap_head 的信息字段中记录 rmap_cmp 的地址信息。后续针对该 1G 内存页中的其它 GPA 建立映射时，在 rmap_cmpt 结构中增加映射计数即可。

2）若标志信息为 rmap_extd 对应的第二值，表示该第二 GPA 对应的最后一级页表项对应的页面粒度为 4k。

a）从 rmap_head 的信息字段中解析出 rmap_extd 结构，即获取 rmap_extd 的

地址信息；

b）根据第二 GPA 在 2m 中 4k 粒度的偏移量 oft，从 val[VAL_EXTD_NR]数组中选取 rmap_extd 中第 oft 个 val 元素，即第二结构体的地址信息；

c）将该第 oft 个 val 元素解析为新的 rmap_head；

d）向该新的 rmap_head 中添加 map 值（即，pte_t 页表项的地址信息），即在第二结构体的信息字段中添加 pte_t 页表项的地址信息。

e）新增 rmap_extd 结构中的映射计数 num。

3）若标志信息为 rmap_head 对应的第一值或者未设置标记信息，则最后一级页表项对应的页面粒度为 2m，不需要解析出下层结构。

a）直接向 rmap_head 的信息字段中添加 map 值，即最后一级页表项（pmd_t）的地址信息。

本实施例中还包括针对第二 RMAP 表增加第二 HPA 对应的 GPA，该过程与第一 RMAP 表增加第二 GPA 对应的 HPA 的实施方式类似，在此不再赘述。

另外，由于存储 HPA 与 GPA 一对多的情况下，在第二 RMAP 表中增加第二 HPA 对应的 GPA 的情况下，还可以包括一对多映射的增加操作，下面重点介绍“一对多”相对于“一对一”不同的部分，其它不再赘述。

以第三结构体中记录第二 HPA 到对应 GPA 的地址转换信息为例进行说明，其中，在第四结构体中记录对应 GPA 的地址转换信息的实施方式相同或相似在此不再赘述。

若是第一次向第三结构体中添加对应 GPA 的地址转换信息，则直接在第三结构体的信息字段中记录第二 HPA 对应 GPA 的地址转换信息。

在第二 HPA 对应多个 GPA 的情况下，判断第三结构体的标志信息中是否存在表示“一对多”的标记信息。若不存在标记信息，表明是第一次扩展第三结构体，可以分配一个第五结构体（rmap_desc）结构体，该 rmap_desc 结构体包括：val[VAL_DESC_NR]，该数组可以初始化为一个特殊值，只要特殊值不被当作地址信息使用即可。例如，ffffffff，f 表示十六进制数中的最大数，即 15，ffffffff 是 4GB 内存空间中的最后一个地址。将第三结构体中记录的 GPA 的地址转换信息，记录到 val[VAL_DESC_NR]中第一个元素的位置，在将第二次要添加的 GPA 的地址转换信息记录在该 val[VAL_DESC_NR]中第二个元素的位置。将该 rmap_desc 结构体的地址信息记录到第三结构体的信息字段中，并在第三结构体中增设表示一对多关系的标志位。

若第三结构体中存在标记信息，则表明已经经过至少一次扩展，则首先在 rmap_desc 中查找是否存在空闲的未记录位置，如果存在则选取该位置记录 map 值，即第二 HPA 对应 GPA 的地址转换信息。若在 rmap_desc 中未找到记录位置，则新分配一个 rmap_desc 并挂载到之前的最后一个 rmap_desc 的 more 指针后面，然后从新分配的 rmap_desc 的数组中选取一个空闲位置记录 map 值，即第二 HPA 对应 GPA 的地址转换信息。

三、针对第一 RMAP 表删除第三 GPA 对应的 HPA。

1、首先根据第二 GPA 的地址信息和第一 RMAP 表中包含的各基础结构体对应的地址空间，从各基础结构体中确定第二 GPA 对应的第一结构体（rmap_head）。其中，可以将第一 RMAP 表看作一个大数组，大数组中的元素为基础结构体，每个基础结构体对应的地址空间为 2m，因此，根据第二 GPA 的地址信息，以 2m 为粒度从大数组中选取第二 GPA 对应的基础结构体，将该选取的基础结构体作为第一结构体。

2、确定第二 GPA 的最后一级页表的页表级别（level）；获取第一结构体的标志信息，判断第一结构体的标志信息指示第一结构体的信息字段中存储的地址信息是否对应于页表级别（level）的结构体，若是，则认为已经存在页表级别（level）的结构体，执行步骤 3。若否，则认为不存在页表级别（level）对应的结构体，也不存在第三 GPA 对应的 HPA，则报错返回。例如，页表级别 level 为 PUD 级别，但是第一结构体的标志信息未设置为 rmap_cmpt 对应的值，则报错返回。

3、根据第一结构体 rmap_head 中信息字段（val）的标志信息选择不同的删除方式。

1）若标志信息为 rmap_cmpt 对应的第二值，表示第二 GPA 的最后一级页表项对应的页面粒度为 1G。

a)从 rmap_head 的信息字段(val)中解析出 rmap_cmpt 结构,即获取 rmap_cmpt 的地址信息；

b）将 rmap_cmpt 结构中的信息字段解析为新的 head_rmap；

c）从新的 rmap_head 中删除 map 值，即 pud_t 的地址信息；

d）减少 rmap_cmpt 结构中的映射计数 num；

e）若映射计数 num 变为 0，则释放 rmap_cmpt 结构。

需要说明的是，在释放 rmap_cmpt 结构时，rmap_cmpt 结构对应 1G 内存页，该 1G 内存页下的 512 个 2m 内存页分别对应的 rmap_head 中的 val 值存储的是 rmap_cmpt 结构的地址信息，若释放 rmap_cmpt 结构，则可以将各 rmap_head 中的 val 值清空。

2）若标志信息为 rmap_extd 对应的第二值，表示该第二 GPA 对应的最后一级页表项对应的页面粒度为 4k。

a）从 head_rmap 的信息字段中解析出 rmap_extd 结构，即获取 rmap_extd 的地址信息；

b）根据第二 GPA 在 2m 中 4k 粒度的偏移量 oft，从 val[VAL_EXTD_NR]数组中选取 rmap_extd 中第 oft 个 val 元素，即第二结构体的地址信息；

c）将该第 oft 个 val 元素解析为新的 rmap_head；

d）从该新的 rmap_head 中删除 map 值，即 pte_t 的地址信息，并将映射计数减 1，计数为 0 的情况下，释放 rmap_extd 结构，同时上一层的 rmap_head 中的映射也

要清空，即清空 rmap_head 的 val 记录的 rmap_extd 的地址信息。

3）若标志信息为 rmap_head 对应的第一值或者未设置标记信息，则最后一级页表项对应的页面粒度为 2m，不需要解析出下层结构。

a）将 rmap_head 中的信息字段中记录的 map 值删除，该 map 值为 pmd_t 的地址信息。

本实施例中还包括针对第二 RMAP 表删除第三 HPA 对应的 GPA，该过程与第一 RMAP 表删除第三 GPA 对应的 HPA 的实施方式类似，在此不再赘述。

另外，由于存储 HPA 与 GPA 一对多的情况下，在第二 RMAP 表中删除第三 HPA 对应的 GPA 的情况下，还可以包括一对多映射的增加操作，下面重点介绍“一对多”相对于“一对一”不同的部分，其它不再赘述。

以第三结构体中删除第三 HPA 到对应 GPA 的地址转换信息为例进行说明，其中，在第四结构体中删除对应 GPA 的地址转换信息的实施方式相同或相似在此不再赘述。

针对第三 HPA 对应多个 GPA 的情况，则从第三结构体的信息字段中获取 rmap_desc 结构体的地址信息，循环 rmap_desc 结构体中记录的各个元素，判断是否是需要删除的第三 HPA 到对应 GPA 的地址转换信息；

若找到需要删除的第三 HPA 到对应 GPA 的地址转换信息（后续简称为 map 值），且存在 rmap_desc 结构，则先查找 rmap_desc 中最后一个存在 map 值的位置 j；

将最后一个位置 j 的值替换到待删除位置 i，并删除位置 j 的值；如果此时 j 不为 0，则返回；若 j 为 0，则表明整个 rmap_desc 为空，可以删除该 rmap_desc；若存在上一个 rmap_desc，则将当前 rmap_desc 的 more 下面挂载的 rmap_desc 信息更新到上一个 rmap_desc 的 more 链表下面去；若不存在上一个 rmap_desc，将 rmap_head 中 val 的值，关联到当前 rmap_desc 的 more 所指向的 rmap_desc。

四、在第一 RMAP 表中将第四 GPA 对应的最后一级页表的第一地址信息修改为第二地址信息。在第二 RMAP 表中将第四 HPA 对应的 GPA 的第一地址转换信息修改为第二地址转换信息。以最后一级页表是 PMD 级别为例进行说明。

1、清理原来 PMD 级别的 HPA-GPA 的映射，即清理第二 RMAP 表中记录的第四 HPA 对应的 GPA 的第一地址转换信息；

2、在 EPT 中建立新的 PMD 级别的页表映射，将新的 HPA 对应的物理页记录在 PMD 级别的页表项中；

3、同时建立新的 PMD 级别的 HPA-GPA 的映射，即将第四 HPA 对应的 GPA 的第二地址转换信息记录在第二 RMAP 表中；

4、刷新 EPT 页表，使得修改生效。

在后续宿主机运行过程中，可能进行页表的扩展、收缩和更新等操作，这时，在更新 EPT 的情况下，可以同步更新第一 RMAP 表和第二 RMAP 表。

五、在 EPT 中将最后一级页表扩展为下一级页表的情况下，在第一 RMAP 表中将最后

一级页表中各GPA各自对应的最后一级页表项的地址信息修改为相应下一级页表的地址信息。

下面以将PMD级别的EPT页表扩展为PTE级别为例进行说明。

1、判断是否需要进行EPT页表的扩展，即EPT页表是否已经是PTE级别；若是，则直接返回；否则继续；

2、分配一个页面EPTP用于记录下一级别PTE级别的页表。EPTP全称为EPT页表指针 (extended-page-table pointer)，EPTP是存放EPT页表物理地址的字段。

3、清理之前的PMD级别的页表映射，包括清理原来PMD页表记录的物理页的地址信息；

4、清理原来第一RMAP表中的GPA_HPA的映射（又称为GPA_EPT映射），以及HPA_GPA的映射；

5、为下一级每个PTE级别的页表，建立对应的映射关系（即在每个PTE级别的页表项中写入物理页的基地址），并将PTE级别的页表项的地址信息记录到分配的页面EPTP中；

6、在每个PTE级别的页面项建立的过程中，还会同步建立新的GPA_HPA的映射和HPA_GPA的映射，分别得到扩展后的第一RMAP表和第二RMAP表。

7、最后将新分配页面EPTP关联到原来PMD级别的页表项中；

8、刷新EPT页表，使得更新生效。

六、在EPT中地址连续的最后一级页表收缩为上一级页表的情况下，在第一RMAP表中将最后一级页表中各GPA各自对应的最后一级页表项的地址信息修改为上一级页表项的地址信息。

下面以将PTE页表收缩为PMD页表为例进行说明。

1、判断是否需要进行EPT页表收缩，即EPT页表是否已经是PMD级别；若是则直接返回；否则继续；

2、判断下一级PTE页表是否可以收缩，即下一级PTE页表的映射关系全为空，或者记录是一段连续的HPA范围，且第一个HPA地址是PMD级别对齐的；

3、为下一级每个PTE级别的页表，清理对应的映射关系，如，清理PTE页表中各页表项记录的物理页的基地址；

4、在每个PTE级别的页面项清理的过程中，还会同步清理旧的GPA_EPT的映射和HPA_GPA的映射，即清理第一RMAP表和第二RMAP表中的相应映射关系；

5、建立新的PMD级别的页表映射，将HPA的映射信息记录在EPT页表项中，解除了原来下级PTE页表的关联；

6、同时还要建立新的PMD级别的RMAP表信息，包括GPA_EPT的映射，以及HPA_GPA的映射；

7、刷新EPT页表，使得更新生效；

8、释放原来记录 PTE 级别页表的页面。

七、还可以包括：清理设定页表级别的页表项，比如该页面别被换出的时候，则更新 HPA_GPA 的映射；

下面以清理 PMD 级别的页表项为例进行说明。

1、清理原来 PMD 级别的 HPA_GPA 的映射；

2、清空记录在 PMD 级别的页表项中的旧的 PMD 级别的页表映射；

3、刷新 EPT 页表，使得更新生效。

本实施例中，除了提供上述地址转换方法之外，还提供一种快速映射表构建方法，如图 1b 所示，该方法包括：

R21、将 EPT 对应的 GPA 地址空间划分为至少一个地址空间，针对至少一个地址空间分别创建基础结构体，基础结构体对应设定页表级别，EPT 包括至少一个 GPA 各自对应的多级页表；

R22、针对 EPT 中任一 GPA，根据任一 GPA 的地址信息和各基础结构体对应的地址空间，从各基础结构体中确定任一 GPA 对应的第一结构体；

R23、根据任一 GPA 对应的最后一级页表的级别与设定页表级别的关系，通过第一结构体记录任一 GPA 对应的最后一级页表项的地址信息，以构建与 EPT 对应的第一 RMAP 表，第一 RMAP 表中存储有至少一个 GPA 各自对应的最后一级页表项的地址信息。

关于本公开实施例提供的图 1a 和图 1b 所示方法中各步骤的详细实施方式以及有益效果已经在前述实施例中进行了详细描述，此处将不做详细阐述说明。

本公开实施例中除了上述方法之外，还包括针对虚拟化层的介绍，如下：

在本公开实施例中，如图 2a 所示，宿主机包括硬件资源，硬件资源上运行有宿主机操作系统（host Operating System，host OS），宿主机操作系统之上运行有各种应用程序；相对于硬件资源，宿主机操作系统和应用程序属于宿主机的软件资源。宿主机操作系统是宿主机中最基本的系统软件，它负责控制和管理整个宿主机的硬件资源和软件资源，并且能够合理地调度宿主机的工作和资源分配，可以提供给用户和其他软件或设备方便的接口和环境。其中，宿主机操作系统可以提供各种系统级的功能，例如文件管理、内存管理、各种输入/输出（Input/Output，IO）设备管理等，基于这些系统功能，宿主机操作系统可以为应用程序提供运行过程中所需的各种资源，以支撑应用程序的运行。

在这些系统功能中，至少部分系统功能，需要在安装宿主机操作系统时进行配置，例如是否开启，开启哪种模式等等。对于未被开启的系统功能，在宿主机操作系统运行过程中将无法使用，由于无法使用该系统功能，会导致依赖该系统功能的应用程序因无法获得该系统功能提供的资源而运行异常或无法运行。但是，随着宿主机操作系统的运行以及应用需求的变化，可能需要使用这些未被开启的系统功能，或者需要对已开启的系统功能进行升级，即宿主机操作系统面临着能力拓展问题。对于传统宿主

机而言，或者重新安装宿主机操作系统，或者在线强行开启部分系统功能。对于重新安装宿主机操作系统的方式需要中断所有应用的运行、效率较低、成本较高；对于在线强行开启相应系统功能的方式，会严重影响系统的稳定性。

在本公开实施例中，为了在不重新安装宿主机操作系统且不影响宿主机操作系统运行性能的情况下，解决宿主机操作系统面临的能力拓展问题，提供了一种新的物理机架构，针对处于运行状态的宿主机操作系统提供两种运行模式，即根模式和非根模式，在硬件资源和宿主机操作系统之间插入虚拟化层，至少在虚拟化层中实现虚拟资源提供功能，并由虚拟化层将正在运行的宿主机操作系统从根模式切换为非根模式，在非根模式下提供至少一种虚拟资源并在虚拟资源上至少运行宿主机操作系统在根模式下因资源不足而无法运行的应用，在无需重新安装操作系统的情况下，可以针对运行态的宿主机操作系统灵活实现虚拟资源提供功能，解决由于未开启或缺失系统功能等原因导致应用程序因资源不足而无法运行的问题，支撑更多应用的运行，有利于扩展宿主机操作系统的能力。

例如，假设宿主机操作系统支持功能 A、功能 B、功能 C 和功能 D 等，在安装宿主机操作系统时，开启了功能 A、功能 B 以及功能 C，功能 D 未开启，且缺失功能 E（即宿主机操作系统不支持功能 E）。采用本公开实施例提供的物理机，在硬件资源和宿主机操作系统之间插入虚拟化层，至少在虚拟化层中实现虚拟资源的提供功能，例如，宿主机操作系统未开启的功能 D 和缺失功能 E，由虚拟化层将宿主机操作系统从根模式切换为非根模式，在非根模式下提供与功能 D 和功能 E 对应的虚拟资源，在该虚拟资源上运行原本因功能 D 未开启和功能 E 缺失而无法运行的应用，在无需重新安装宿主机操作系统的情况下，可以针对运行态的宿主机操作系统灵活实现虚拟资源的提供功能，解决因未开启功能 D 和缺失功能 E 而导致相应应用无法运行的问题，有利于扩展宿主机操作系统的能力。进一步，为了实现对宿主机操作系统不足的弥补，在非根模式下可以优先提供与功能 D 和功能 E 对应的虚拟资源，解决依赖于功能 D 和功能 E 的应用的运行问题；当然，为了宿主机操作系统的兼容性，避免模式的频繁切换，还可以在非根模式下提供与功能 A、功能 B 和功能 C 等对应的虚拟资源，以便于在这些资源上运行依赖于功能 A、功能 B 和功能 C 的相应应用。

在此说明，在本公开实施例中，重点关注宿主机操作系统在两种模式之间的切换过程，至于在非根模式中提供了与哪些系统功能对应的虚拟资源以及是如何提供与这些系统功能对应的虚拟资源的不做限定。以下结合附图，对本公开实施例提供的物理机架构和宿主机操作系统从根模式切换到非根模式的过程进行详细说明。

图 2b 为本公开示例性实施例提供的一种物理机的结构示意图。如图 2b 所示，该物理机包括：硬件资源 10 和运行在硬件资源上的宿主机操作系统 20，在硬件资源 10 和宿主机操作系统 20 之间实现有虚拟化层 30。进一步，在宿主机操作系统 20 之上还包括应用层 40。应用层 40 中包括各种应用程序。

可选地,硬件资源 10 包括:至少一个物理计算资源对象 101 和物理存储介质 102,物理存储介质 102 负责提供物理地址空间。除此之外,物理机上的硬件资源 10 还可以包括:IO 设备、通信组件、显示器、电源组件以及音频组件等其它组件,未作图示。

其中,物理计算资源对象 101 可以是中央处理器(Central Processing Unit,CPU)、图形处理器(Graphics Processing Unit,GPU)、数据处理器(Data Processing Unit,DPU)、张量处理器(Tensor Processing Unit,TPU)、云基础设施处理器(Cloud Infrastructure Processing Unit,CIPU)以及专用集成电路(Application Specific Integrated Circuit,ASIC)等各种具有计算能力的物理资源对象。宿主机操作系统 20 可以运行在至少一个物理计算资源对象 101 上。

其中,物理存储介质 102 包括永久性和非永久性、可移动和非可移动媒体可以由任何方法或技术来实现信息存储。信息可以是计算机可读指令、数据结构、程序的模块或其他数据。物理存储介质 102 包括但不限于:相变内存(Phase-change Random Access Memory,PRAM)、静态随机存取存储器 SRAM)、动态随机存取存储器(Dynamic Random Access Memory,DRAM)、其他类型的随机存取存储器(RAM)、只读存储器(ROM)、电可擦除可编程只读存储器(EEPROM)、快闪记忆体或其他内存技术、只读光盘只读存储器(CD-ROM)、数字多功能光盘(Digital Video Disc,DVD) 或其他光学存储、磁盒式磁带,磁带磁盘存储或其他磁性存储设备或任何其他非传输介质,可用于存储可以被计算设备访问的信息。

其中,物理内存是物理机中实际存在的存储器芯片或模块,用于存储数据和程序代码。物理内存可能包括物理存储介质 102 中的非永久性存储器,随机存取存储器(Random Access Memory,RAM)和/或非易失性内存等形式,如只读存储器(ROM)或闪存(flash RAM)。

其中,物理地址空间是物理内存对应的地址范围。物理地址空间通常与硬件架构和宿主机操作系统相关,其描述了实际内存条的物理位置和大小。在虚拟内存系统中,物理地址空间还涉及到内存页表等机制,内存页表机制用于实现客户机物理地址到宿主机物理地址的映射。

在本实施例中,宿主机操作系统 20 对应有根模式和非根模式。根(root)模式是宿主机操作系统 20 对硬件资源 10 具有直接访问权限的工作模式,非根模式(non-root)是由虚拟化层 30 对硬件资源 10 进行虚拟化并代替宿主机操作系统 20 对虚拟化后的资源进行调度和访问的工作模式。在非根模式下虚拟化层 30 对硬件资源 10 的访问是受限的。相应地,宿主机操作系统 20 在非根模式下也可以被称为客户机操作系统。

在本实施例中,在硬件资源 10 和宿主机操作系统 20 之间插入一种轻量级、功能强大且高效的虚拟化层 30。虚拟化层 30 可以预先开发并插入硬件资源 10 和宿主机操作系统 20 之间。其中,插入虚拟化层 30 的方式并不限定。例如,可以在物理机上安装宿主机操作系统 20 之后,宿主机操作系统 20 处于正常运行的状态下插入虚拟化层

30。

其中，虚拟化层 30 具有虚拟资源提供能力，至少能够提供与宿主机操作系统 20 在根模式下因未开启或缺失的系统功能而无法提供的资源对应的虚拟资源。也就是说，宿主机操作系统 20 在根模式下，由于未开启或缺失的系统功能而无法提供的资源，可以由虚拟化层 30 提供对应的虚拟资源，以便于运行相应应用。具体地，虚拟化层 30 可以将运行中的宿主机操作系统 20 从根模式切换到非根模式，并在非根模式下提供至少一种虚拟资源，以及在至少一种虚拟资源上运行至少一种应用，在无需重新安装操作系统的情况下，可以针对运行态的宿主机操作系统灵活实现虚拟资源提供功能，解决宿主机操作系统 20 在根模式下因未开启或缺失的系统功能而无法提供相应资源运行相应应用的问题，可以支撑更多应用的运行，有利于扩展宿主机操作系统的能力。进一步，出于宿主机操作系统兼容性的考虑，减少宿主机操作系统频繁的模式切换，在非根模式下，不仅可以针对宿主机操作系统 20 在根模式下未开启或缺失的系统功能而提供相应的虚拟资源，也可以针对宿主机操作系统 20 在根模式下已开启的系统功能而提供相应虚拟资源，实现对宿主机操作系统 20 更多功能的支持。

举例说明，在虚拟化层 30 中实现的至少一种目标功能包括但不限于：内存管理功能、文件管理功能、虚拟化功能、调度优化功能等。这里的内存管理功能、文件管理功能、调度优化功能等可以是新拓展的宿主机操作系统原本不支持的功能，也可以是宿主机操作系统原本支持但在安装宿主机操作系统过程中未开启的功能。无论是新拓展的还是未开启的功能，内存管理功能具体包括但不限于：内存缺页管理、内存交换管理、内存映射管理等；文件管理功能具体包括但不限于：目录管理、权限管理、文件的备份、创建、查询、删除、修改等；虚拟化功能包括但不限于：对各种弹性资源（如 CPU、GPU、内存、网络等）的虚拟化等；调度调优功能包括但不限于：对物理计算资源对象的调度以及各种虚拟化资源的调度等。

进一步，图 2b 所示为物理机的整体架构示意图，相对图 2b 所示，图 2c 和图 2d 示出了工作在根模式和非根模式下的物理机的内部架构。在进行模式切换之前，宿主机操作系统工作在根模式下，在该根模式下，如图 2c 所示，运行中的物理机的内部架构自下而上为硬件资源 10、宿主机操作系统 20 和应用层 40；在进行模式切换之后，工作在非根模式，在该非根模式下，如图 2d 所示，运行中的物理机的架构自下而上为硬件资源 10、虚拟化层 30、客户机操作系统和应用层 40。其中，客户机操作系统是对宿主机操作系统 20 运行在非根模式下时的称谓。

在本公开实施例中，针对处于运行状态的宿主机操作系统提供两种运行模式，即根模式和非根模式，在硬件资源和宿主机操作系统之间插入虚拟化层，至少在虚拟化层中实现虚拟资源的提供功能，并由虚拟化层将正在运行的宿主机操作系统从根模式切换为非根模式，在非根模式下提供虚拟资源并在虚拟资源上至少运行宿主机操作系统在根模式下因资源不足而无法运行的应用，在无需重新安装宿主机操作系统的情况

下，可以针对运行态的宿主机操作系统灵活实现虚拟资源提供功能，支撑更多应用的运行，扩展宿主机操作系统的能力。

在此说明，从程序代码的角度来看，本公开实施例中涉及的“提供虚拟资源的功能”以及“从根模式切换到非根模式”等实现代码属于虚拟化层 30，这些代码位于宿主机操作系统下；但是，从运行态的角度来看，在真正完成模式切换之前，虚拟化层 30 中的至少部分程序代码是在宿主机操作系统的运行过程中被执行的，换句话说，虚拟化层 30 中的至少部分程序代码是在根模式下被执行的，其中，虚拟化层 30 中在根模式下执行的程序代码至少包括：用于从根模式切换到非根模式的程序代码，例如下面实施例中提到的切换函数中的上半部分等。相应地，在真正完成模式切换之后，部分程序代码会在非根模式下执行，例如下面实施例中提到的切换函数的下半部分等。另外，虚拟化层 30 中负责提供虚拟资源的程序代码也是运行在根模式下，但是这部分代码是在切换完成之后被运行的；虚拟化层 30 还会对外提供一些接口代码，这些接口代码部分会在非根模式下运行，部分需要切换到根模式下运行，具体视接口代码的功能而定。

在一可选实施例中，如图 3a 所示，虚拟化层 30 将运行中的宿主机操作系统从根模式切换到非根模式包括三部分内容：

S1、创建内存页表，以用于存储非根模式下客户机物理地址与根模式下宿主机物理地址之间的映射关系；

S2、创建模式切换时所需的信息承载对象，信息承载对象用于在根模式与非根模式之间实现上下文信息的同步；

S3、基于信息承载对象和内存页表，将运行中的宿主机操作系统从根模式切换到非根模式。

下面对上述三部分内容的详细实现进行说明。

步骤 S1：创建内存页表

在根模式下物理地址空间具有宿主机物理地址，宿主机物理地址可以用 HPA（Host Physical Address）表示，宿主机物理地址是物理内存资源中物理内存的地址；在非根模式下，物理地址空间对应有虚拟地址空间，虚拟地址空间具有客户机物理地址，客户机物理地址可以用 GPA（Guest Physical Address）表示，客户机物理地址是虚拟内存资源中虚拟内存的地址。

在本实施例中，虚拟化层 30 可以创建内存页表，内存页表用于存储非根模式下的客户机物理地址与根模式下的宿主机物理地址之间的映射关系。内存页表可以用 EPT（Extended Page Tables）表示。需要说明的是，在初始建立内存页表时，由于宿主机操作系统还未在非根模式中运行，也没有执行内存回收或内存交换等操作，则非根模式下的客户机物理地址与根模式下的宿主机物理地址是相同的。后续随着宿主机操作系统在非根模式下的使用，例如，内存回收或内存交换等，客户机物理地址可能不再

等于宿主机物理地址，内存页表可以动态维护客户机物理地址与宿主机物理地址之间的映射关系。

在一可选实施例中，如图 3b 所示，内存页表的创建过程包括：S11、建立内存页表对应的页表结构体；S12、从物理地址空间中申请内存页表的根页面；S13、根据根页面的宿主机物理地址生成内存页表的入口地址，并添加到页表结构体中；S14、根据物理地址空间的地址范围和用于对客户机物理地址与宿主机物理地址进行映射的页面粒度，创建多级内存页表。

首先创建页表结构体，该页表结构体的作用是存储与内存页表相关的各种信息，例如存储内存页表的入口地址、操作内存页表的地址锁数组（Address Lock Array），以及特殊的高级可编程中断控制器（Advanced Programmable Interrupt Controller，APIC）页面的地址等。其中，内存页表的入口地址用于访问内存页表，例如，入口地址可以是内存页表的首地址。地址锁数组中每个条目都对应一个页表条目，并包含一个锁标志位，用于表示该条目是否被锁定，在未被锁定的情况下，进程可以访问该页表条目对应的内存空间。特殊的 APIC 页面的地址，主要用于访问特殊的 APIC 页面。在创建页表结构体之后，可以从物理存储介质提供的物理地址空间中申请一个物理页面，作为内存页表的根页面；根据该根页面的宿主机物理地址生成内存页表的入口地址，并添加到页表结构体中。其中，根据物理计算资源对象的类型以及生产厂商的不同，内存页表的入口地址的形式也有所不同。例如，可以直接将根页面的宿主机物理地址作为内存页表的入口地址。又例如，可以是将根页面的宿主机物理地址加上设定的偏移量或者标志位，作为内存页表的入口地址，对此不作限定。

其中，物理地址空间具有地址范围，例如，物理地址空间的地址范围可以是 0-8G 或 0-16G 等。另外，在本实施例中，在非根模式下，可以打破宿主机操作系统所支持的单一页面粒度，允许按需设定进行地址映射的页面粒度，例如地址映射的页面粒度可以包含但不限于：4K（Kilobyte，千字节）、2M（Megabyte，兆字节）或 1G（Gigabyte，吉字节）等。鉴于此，在一可选实施例中，虚拟化层 30 可以根据物理地址空间的地址范围和用于对客户机物理地址与宿主机物理地址进行映射的页面粒度，创建多级内存页表。其中，多级内存页表可以是 2 级内存页表、4 级内存页表或 5 级内存页表等，具体根据物理地址空间的地址范围和页面粒度而定。例如，针对相同的地址范围，页面粒度越小，所需的内存页表层级越多；又例如，针对相同的页面粒度，地址范围越大，所需的内存页表层级越多。

可选地，在本实施例中，将物理地址空间划分为第一地址空间和第二地址空间。第二地址空间大于第一地址空间，也就是说，第二地址空间的地址大小大于第一地址空间的地址大小。例如，在物理地址空间的大小为 8G 的情况下，可以将物理地址空间的地址范围划分为 0G-4G 以及 4G-8G，第一地址空间可以是 0G-4G 对应的地址空间，第二地址空间可以是 4G 以上的地址空间；又例如，若物理地址空间的大小为 16G，

可以将物理地址空间划分为 0G-4G 以及 4G-16G，第一地址空间可以是 0G-4G 对应的地址空间，第二地址空间可以是 4G 以上的地址空间。基于此，在创建多级内存页表过程中，可以针对第一地址空间和第二地址空间分别创建各自对应的多级内存页表。其中，第一地址空间主要用作内存地址空间，第二地址空间至少可以用作 IO 地址空间。其中，内存地址空间用于描述程序和数据在内存中的存储位置，IO 地址空间用于描述物理机的 I/O 设备的地址范围。例如，物理地址空间通常被分为几个不同的地址范围。将地址范围是 0-4GB 的地址空间称为第一地址空间，将地址范围是 4GB 以上的地址空间称为第二地址空间。第一地址空间通常被称为“基本内存”或“系统内存”，其包括宿主机操作系统、应用程序和基本硬件驱动程序所使用的内存，即第一地址空间主要用作内存地址空间。第二地址空间通常被称为“大容量内存”或“扩展内存”。这种内存容量通常用于高性能计算机、服务器和工作站等设备，以支持更复杂的计算任务和多任务处理。例如，第二地址空间至少可以用作 IO 地址空间。

其中，针对第一地址空间，以第一页面粒度形成客户机物理地址与宿主机物理地址之间的第一映射关系，即客户机物理地址与宿主机物理地址之间按照第一页面粒度的页面大小进行映射，根据第一映射关系创建第一地址空间对应的多级内存页表；针对第二地址空间中的 IO 地址空间，以第二页面粒度形成客户机物理地址与宿主机物理地址之间的第二映射关系，即客户机物理地址与宿主机物理地址之间按照第二页面粒度的页面大小进行映射，根据第二映射关系创建 IO 地址空间对应的多级内存页表；第二页面粒度大于第一页面粒度，例如，第一页面粒度为 2M，第二页面粒度为 1G。其中，采用第一页面粒度（小页面粒度）进行内存映射，可以减小内存页面中的内部碎片，减少内存浪费，并减少缓存中页表项的丢失，提高地址转换效率；采用第二页面粒度（大页面粒度）进行内存映射，减少页表项数据，减少内存页表的层级，降低内存管理开销，提高物理计算资源对象的性能。

进一步可选地，第二地址空间还可以用作内存地址空间。基于此，虚拟化层 30 还可以针对第二地址空间中的内存地址空间，以第三页面粒度形成客户机物理地址与宿主机物理地址之间的第三映射关系，即客户机物理地址与宿主机物理地址之间按照第三页面粒度的页面大小进行映射，根据第三映射关系创建内存地址空间对应的多级内存页表；其中，第三页面粒度小于第二页面粒度，但对第三页面粒度的大小不做限定。

在一可选实施例中，第三页面粒度可以等于第一页面粒度，例如，第一页面粒度和第三页面粒度均为 2M，也就是说，针对内存地址空间，可以采用较小的页面粒度进行地址空间映射，对于 IO 地址空间，可以采用较大的页面粒度进行地址空间映射，可以合理的利用物理地址空间，提高内存管理的性能。

进一步可选地，在一些应用场景中，第二地址空间中的内存地址空间中包括 APIC 页面， APIC 页面采用更小的页面粒度，例如第四页面粒度，第三页面粒度大于第四

页面粒度。例如，第三页面粒度为 2M，第四页面粒度为 4K。

在第二地址空间中的内存地址空间包含第四页面粒度的 APIC 页面的情况下，意味着该第四页面粒度的 APIC 页面属于某个第三页面粒度的物理页，将第四页面粒度的 APIC 页面所属的第三页面粒度的物理页简称为第一物理页，则可以获取 APIC 页面所属的第三页面粒度的第一物理页面以及第一物理页面对应的目标内存页表，其中，第一物理页面对应的目标内存页表是多级内存页表中的最后一级页表，该页表中的页表项指向第一物理页面中的地址空间；为了适配 APIC 页面的页面粒度，可以将第一物理页面拆分为第四页面粒度的多个子页面，并对目标内存页表下面继续扩展下一级内存页表，即在目标内存页表下添加下一级内存页表，下一级内存页表用于存储拆分出的多个子页面的宿主机物理地址，例如，在下一级内存页表的每个页表项中分别存储一个子页面对应的宿主机物理地址，该宿主机物理地址为子页面的首地址。

在本实施例中，在创建内存页表时，采用第一页面粒度、第二页面粒度、第三页面粒度以及第四页面粒度相结合的混合映射粒度的方式，不仅支持小页粒度，还支持大页粒度，可以满足不同物理页映射的需求，提高内存映射的灵活度。

进一步，在本公开实施例提供的非根模式下，在创建内存页表时，不再对宿主机的物理地址空间进行限定，允许为全量物理地址空间创建内存页表即支持全量内存映射，因此既能够针对用户态页面创建内存页表，也支持针对内核态页面创建内存页表，通过支持全量物理地址空间的内存映射，为后续基于内存映射的各种操作提供基础框架，例如在基于内存映射进行内存交换时，不仅支持对用户态页面的交换，还支持对内核态页面的交换，即支持全量内存交换。

在本实施例中，以第一页面粒度是 2M，第二页面粒度是 1G，第三页面粒度是 2M，第四页面粒度是 4K，多级内存页表实现为四级内存页表为例，对创建内存页表的具体过程进行示例性说明。

下面给出四级内存页表的一种示例，四级内存页表包括：页全局目录（Page Global Directory，PGD）、页上级目录（Page Upper Directory，PUD）、页中间目录（Page Middle Directory，PMD）以及页表（Page Table Entry，PTE）。其中，PGD 最高级别的目录，PGD 中包括多个全局页目录项 pgd_t，每个 pgd_t 可以映射 512G 大小的宿主机物理地址（HPA），pgd_t 指向下一级页目录（PUD）；PUD 中包括多个上级页目录项 pud_t，每个 pud_t 可以映射 1G 大小的宿主机物理地址（HPA），pud_t 指向下一级页目录（PMD）；PMD 中包括多个中间页目录项 pmd_t，每个 pmd_t 可以映射 2M 大小的宿主机物理地址（HPA），pmd_t 指向下一级页目录（PTE），PTE 中包括多个直接页目录项 pte_t，每个 pte_t 可以映射 4K 大小的宿主机物理地址（HPA），每个 pte_t 指向宿主机物理地址（HPA）中对应的物理页。

1）针对地址范围是 0G-4G 的第一地址空间，以 2M 的页面粒度形成客户机物理地址与宿主机物理地址之间的第一映射关系，根据第一映射关系创建第一地址空间对

应的多级内存页表。

a1、根据第一地址空间中的客户机物理地址（GPA），根据客户机物理地址（GPA）对应 PGD 的索引值，确定 PGD 级别页表中的页表项 pgd_t，该 pgd_t 指向 PUD 级别页表。

b1、根据 pgd_t 中的标志位，判断 PUD 级别的页表是否存在，若不存在，则申请一个页面作为 PUD 级别页表。

c1、若存在 PUD 级别页表，则根据第一页面粒度（如，2M），判断 PUD 级别页表是否需要做扩展，由于每个 pud_t 可以映射 1G 大小的宿主机物理地址（HPA），1G 大于第一页面粒度（如，2M），则确定需要扩展出下一级 PMD 级别页表，为后续映射做准备。

d1、根据客户机物理地址（GPA）对应 PUD 的索引值，确定 PUD 级别页表中的页表项 pud_t，该 pud_t 指向 PMD 级别页表。

e1、根据该 pud_t 中的标志位，判断下一级 PMD 级别页表是否存在，若不存在，则申请一个页面作为 PMD 级别页表。

f1、根据第一页面粒度（如，2M），判断 PMD 级别页表是否需要扩展，由于每个 pmd_t 可以映射 2M 大小的宿主机物理地址（HPA），则确定不需要扩展出下一级 PTE 级别页表。

g1、根据客户机物理地址（GPA）对应 PMD 的索引值，确定 PMD 级别页表中的页表项 pmd_t。

h1、在映射的页面粒度为 2m 的情况下，直接在 PMD 级别页表中建立页表项 pmd_t，pmd_t 指向宿主机物理地址（HPA）的物理页，物理页的大小为 2M。因此，需要组合出 PMD 级别页表的页表项 pmd_t，pmd_t 可以包括相应 HPA 对应的物理页帧号（Page Frame Number，PFN），并将该页表项设置为可访问，以及将大页标志位置 1。例如，大页可以是大小为 2M 或 1G 等物理页，大小为 4K 的物理页不属于大页。

i1、将组合出的 PMD 级别页表的页表项 pmd_t，填入 PMD 级别页表中，完成一次第一页面粒度（如，2M）的映射，并记录映射完成进度和下一个需要映射的客户机物理地址，直至针对第一地址空间，建立第一页面粒度的映射。

j1、判断第一页面粒度（如，2M）的映射是否完成，未完成回到步骤 b，否则结束。

2）针对第二地址空间中的输入/输出（IO）地址空间，以第二页面粒度形成客户机物理地址与宿主机物理地址之间的第二映射关系，根据第二映射关系创建 IO 地址空间对应的多级内存页表。即根据 IO 内存资源（iomem_resource）中记录的宿主机操作系统的地址空间（即物理内存空间和物理 IO 空间），为第二地址空间中 IO 地址空间（如，4G 以上的内存地址空间）建立第二页面粒度（如，1G 粒度）的多级内存页表。其中，建立多级内存页表的过程类似于上述以第一页面粒度建立多级内存页表的实施

方式，区别在于：建立第二页面粒度（如，1G 粒度）的多级内存页表，需要在 PUD 级别页表中建立 pud_t，pud_t 指向宿主机物理地址（HPA）的物理页，物理页的大小为 1G。

3）针对第二地址空间中的内存地址空间，以第三页面粒度（如，2M）形成客户机物理地址与宿主机物理地址之间的第三映射关系，根据第三映射关系创建内存地址空间对应的多级内存页表；其中，第三页面粒度小于第二页面粒度。其中，建立多级内存页表的过程可参见前述以第一页面粒度建立多级内存页表的实施方式，在此不再赘述。

4）在内存地址空间包含第四页面粒度的 APIC 页面的情况下，获取 APIC 页面所属的第三页面粒度的第一物理页面以及第一物理页面对应的目标内存页表；将第一物理页面拆分为第四页面粒度的多个子页面，并在目标内存页表下添加下一级内存页表，下一级内存页表用于存储拆分出的多个子页面的宿主机物理地址。

5）申请一个特殊的 APIC 页面，APIC 页面的物理地址为客户机物理地址。

6）建立 APIC 的宿主机物理地址和这个特殊 APIC 页面的第四页面粒度（如，4k 粒度）的映射。APIC 默认的宿主机物理基址（APIC_DEFAULT_PHYS_BASE）为（0xfee00000），该物理地址对应于第二内存空间中的内存地址空间，这个过程中涉及到已经建立好的第三页面粒度（如，2M 粒度）页面的拆分，映射过程如下：

a2、依次执行步骤 1）中的步骤 a1-e1，根据第四页面粒度（如，4K），判断 PMD 级别页表是否需要扩展，由于每个 pmd_t 可以映射 2M 大小的宿主机物理地址（HPA），而 2M 不等于 4K，则确定需要扩展出下一级 PTE 级别页表。

b2、根据客户机物理地址（GPA）对应 PTE 的索引值，确定 PTE 级别页表中的页表项 pte_t，组合出 pte_t 的内容，包括 HPA 对应的物理页帧号，并将该页表项设置为可访问，这时，大页标志位不置 1。

c2、将组合出的 PTE 级别页表的页表项内容，填入相应 pte_t 中，完成本次映射。

S2、创建模式切换时所需的信息承载对象

在本实施例中，为了针对宿主机操作系统进行模式切换，一方面针对任一物理计算资源对象 101，创建物理描述符结构体，如图 3c 中的步骤 S21。后续一些描述中可以将物理描述符结构体简称为 pcpu，该物理描述符结构体用于在模式切换时保存任一物理计算资源对象运行在根模式下的上下文信息。任一物理计算资源对象在根模式下的上下文信息可以包含但不限于：任一物理计算资源对象使用到的各种寄存器的值、任一物理计算资源对象的状态信息、宿主机的状态信息和相关配置信息等。其中，相关配置信息包括但不限于：高压（High Voltage，hv）定时器的频率等。

在本实施例中，并不限定物理描述符结构体的实现方式，凡是可以保存物理计算资源对象运行在根模式下的上下文信息的实现结构均适用于本公开实施例。在一可选实施例中，物理描述符结构体可以包含但不限于以下部分或字段：第一寄存器结构体（regs）、承载

物理计算资源对象的状态信息的字段、承载宿主机的状态信息的字段和承载配置信息的字段。第一寄存器结构体可以包含但不限于：通用寄存器字段、调试寄存器字段、栈相关寄存器字段、浮点(Floating Point Unit，fpu)寄存器字段、模型专用寄存器(Model Specific Register，MSR）字段等。

为了针对宿主机操作系统进行模式切换，另一面针对任一物理计算资源对象，创建虚拟化描述符结构体，如图 3c 中的步骤 S22。后续部分描述中可以将虚拟化描述符结构体简称为 vcpu，虚拟化描述符结构体用于在模式切换时同步物理描述符结构体中的上下文信息。该虚拟化描述符结构体代表了对物理计算资源对象进行虚拟化得到的虚拟计算资源对象。

其中，并不限定虚拟化描述符结构体的实现方式，凡是可以同步物理计算资源对象中的上下文信息的实现结构均适用于本公开实施例。在一可选实施例中，vcpu 可以包含但不限于以下字段：与第一寄存器结构体对应的第二寄存器结构体（regs），除此之外，vcpu 还包括:承载虚拟计算资源对象识别指令(如,cpu_id)的字段、承载中断向量号(apic_id)的字段、承载 vcpu 的运行状态的字段、承载 vcpu 切换过程中的状态变化的切换标记的字段、承载 APIC 寄存器的相关配置信息的字段、承载特定模型寄存器（如，MSR 寄存器）的配置信息的字段、承载处理器操作码（如，CPUID）的配置信息的字段、承载中断（Program Interruption，PI）的描述符的字段，以及承载退出事件相关配置信息的字段等。

进一步，为了针对宿主机操作系统进行模式切换，针对任一物理计算资源对象，创建虚拟化控制结构体，如图 3c 中的步骤 S23。例如，虚拟化控制结构体可以实现为 vmcs（Virtual Machine Control System），虚拟化控制结构体用于保存任一物理计算资源对象在非根模式下的运行态信息和运行控制信息。例如，物理计算资源对象在非根模式下的运行态信息可以是一些寄存器的值，以及物理计算资源对象所处的状态，例如，活跃、停机（HLT）或关机（shutdown）等状态。运行控制信息主要是指非根模式下对宿主机操作系统的控制信息，例如，虚拟化控制结构体保存的运行控制信息可以包含但不限于以下字段：承载非根模式的退出事件的字段、承载特定处理器模型寄存器位图（msr_bitmap）的字段、承载 APIC 虚拟化的字段、承载中断虚拟化的字段、承载抢占定时器的字段、承载时钟配置的字段或承载 IO 控制信息的字段等，可以根据实际情况在初始化虚拟化控制结构体时需要按需配置相应字段中的信息。

在一可选实施例中，虚拟化层 30 还可以创建 IO 位图（io_bitmap），如图 3c 中的步骤 S24。IO 位图用于记录任一物理计算资源对象在非根模式下对各个 IO 端口的访问权限，访问权限与运行控制信息中的 IO 控制信息相关。其中，访问权限可以默认都为 0，即允许所有 IO 端口的访问，后续若需要截获部分 IO，可按需配置 io_bitmap。

进一步，虚拟化层 30 还可以创建新栈，为便于描述和区分，将该新栈称为第一栈，如图 3c 中的步骤 S25。新栈（即第一栈）是相对旧栈而言的，为了便于描述和区分，将旧栈称为第二栈，旧栈（即第二栈）是任一物理计算资源对象当前使用的栈，在从根模式切换到非根模式后，任一物理计算资源对象会继续使用旧栈在非根模式下运行。

在此说明，从根模式切换到非根模式后，物理计算资源对象可实现为虚拟物理计算资源对象，例如，物理 cpu 可实现为虚拟 cpu。第一栈是为虚拟化层 30 准备的栈，用于供虚拟化层 30 在根模式基于新栈（即第一栈）进行内存访问。

在一可选实施例中，虚拟化层 30 还可以对虚拟化描述符结构体和虚拟化控制结构体分别进行初始化，如图 3c 中的步骤 S26。初始化的过程是使能所需的功能、为所需功能申请内存页面以及配置初始值的过程。其中，对虚拟化描述符结构体的初始化过程，可以是将需要使能的功能对应的初始化信息添加至虚拟化描述符结构体的相应字段中。对虚拟化控制结构体的初始化过程，可以是将需要使能的功能对应的初始化信息添加至虚拟化控制结构体的相应字段中。

可选地，虚拟化层 30 可以在虚拟化控制结构体中，配置任一物理计算资源对象在非根模式下运行时的运行控制信息。例如，运行控制信息可以包括但不限于：MSR 控制信息、PI 控制信息、APIC 控制信息、硬件寄存器操作指令信息、内存页表的入口地址和退出非根模式的相关配置信息中的至少一种。

进一步可选地，虚拟化层 30 在配置运行控制信息时，具体用于执行以下至少一种配置操作：

a3、建立特定模型寄存器位图（如，msr_bitmap），并分配保存该位图的第二物理页面；

b3、设置 PI 中断功能，并设置 PI 通知向量；

c3、设置 APIC 中断控制功能，从物理地址空间中分配保存 APIC 表的第三物理页面，并在 APIC 字段中保存 APIC 表所在页面的基地址；另外，还需要设置 APIC_ID 为当前物理计算资源对象（如，CPU）的 apic_id，APIC 本地向量寄存器（APIC Local Vector Register，APIC_LVR）继承当前物理计算资源对象（如，CPU）的设置；若截获 apic 定时器还可以配置定时器的配置以及处理函数；

d3、设置处理器操作码（如，CPUID），建立虚拟计算资源对象的缓存；

e3、设置暂停循环退出（PAUSE-Loop Exiting，PLE），PLE 配置用于减少因为循环等待而造成虚拟计算资源的浪费。

f3、将步骤 S1 中生成的多级内存页的入口地址保存在虚拟化控制结构体中，同时还需要设置非根模式退出后执行的 RIP 指令位置；

g3、默认的根模式和非根模式下的通用寄存器的配置。

其中，对每个虚拟化描述符结构体和虚拟化控制结构体进行初始化的过程，如下：

1）依次加载每个虚拟化描述符结构体（vcpu），对每个 vcpu 结构体进行初始化设置；

2）载入 vcpu 对应的虚拟化控制结构体（vmcs）结构作为当前 vmcs；

3）针对当前 vmcs 配置运行控制信息，即执行上述步骤 a3 至步骤 g3；

4）清理当前 vmcs 结构；

5）判断需要初始化的 vcpu 是否初始化完毕，若未初始化完毕，则重复步骤 1）至步骤 4）；若初始化完毕，则结束初始化。

综上可知，针对宿主机操作系统进行模式切换，涉及物理计算资源对象到虚拟计算资源对象的切换，另外涉及上下文信息的同步，IO 操作的访问控制以及内存访问的切换，因此，创建物理计算资源对象对应的物理描述符结构体和虚拟计算资源对象对应的虚拟化描述符结构体，以便于在模式切换过程中负责上下文信息的同步，并创建 IO 位图，以便于在模式切换过程中进行 IO 操作的访问控制，以及创建第一栈，以便于虚拟化层 30 在模式切换过程中进行内存访问的切换。

其中，在创建出内存页表并创建出各种信息承载对象之后，就意味着做好了模式切换的基础，在此基础上，可以基于信息承载对象和内存页表，将运行中的宿主机操作系统从根模式切换到非根模式。

S3、基于信息承载对象和内存页表，将运行中的宿主机操作系统从根模式切换到非根模式

在本实施例中，可以针对任一物理计算资源对象进行模式切换，且针对各个物理计算资源对象进行模式切换的过程相同。为了便于描述和区分，下面以针对目标物理计算资源对象进行模式切换为例进行说明，目标物理计算资源对象可以是任一物理计算资源对象。

如图 3d 所示，针对目标物理计算资源对象进行模式切换的过程包括：步骤 S31：在对目标物理计算资源对象进行模式切换的情况下，将目标物理计算资源对象在根模式下的上下文信息，保存至目标物理计算资源对象对应的物理描述符结构体中。步骤 S32：将物理描述符结构体中的上下文信息中各寄存器的值和段寄存器的配置信息，分别同步至目标物理计算资源对象对应的虚拟化描述符结构体和虚拟化控制结构体中。步骤 S33：根据虚拟化描述符结构体、虚拟化控制结构体以及内存页表，控制目标物理计算资源对象运行，以将运行中的宿主机操作系统从根模式切换到非根模式。

例如，将物理描述符结构体中的上下文信息中各寄存器的值同步至目标物理计算资源对象对应的虚拟化描述符结构体，将物理描述符结构体中的上下文信息中段寄存器的配置信息同步至目标物理计算资源对象对应的虚拟化控制结构体中。其中，段寄存器的配置信息可以包括但不限于：段的大小、段的起始地址和段的管理属性，例如，段的管理属性包括：禁止写入、禁止执行或系统专用等。

在本实施例中，步骤 S1 和步骤 S2 可以在虚拟化层 30 启动或初始化时执行，步骤 S3 可以根据动态需求（外部指令），按需执行。其中，是否对目标物理计算资源对象进行模式切换，可由外部指令而定。

在一可选实施例中，本实施例的宿主机操作系统 20 对外提供一种对称多处理（Symmetrical Multi-Processing，SMP）调用接口，允许外部针对物理机上的任一物理计算资源对象发送 SMP 调用请求，该 SMP 调用请求用于指示需要对该任一物理计算

资源对象进行模式切换。为了便于区分和描述，以外部针对目标物理计算资源对象发送 SMP 调用请求为例进行说明。例如，运行在宿主机操作系统 20 上的上层应用可以向目标物理计算资源对象发起 SMP 调用请求，或者，开发人员可以通过宿主机操作系统 20 向目标物理计算资源对象发起 SMP 调用请求。

目标物理计算资源对象在接收到 SMP 调用请求的情况下，可以发起对切换函数的调用请求，以触发模式切换。相应地，虚拟化层 30 提供用于供宿主机操作系统进行模式切换的切换函数（如，switch_vcpu 函数），用于供目标物理计算资源对象运行切换函数，以将宿主机操作系统从根模式切换至非根模式，具体是指执行将目标物理计算资源对象在根模式下的上下文信息保存至目标物理计算资源对象对应的物理描述符结构体中的操作及后续操作（如，步骤 S32 和 S33），以对物理计算资源对象进行模式切换。在进入非根模式之后，虚拟化层 30 还可以在发生虚拟化控制结构体中配置的退出事件时，控制目标物理计算资源对象从非根模式退出至根模式。

在一可选实施例中，在模式切换过程中，可以在保存上下文信息之前，判断目标物理计算资源对象对应的切换标记的状态是否为待切换状态；在判断结果为是的情况下，将切换标记更新为切换中状态。例如，目标物理计算资源对象可以获取切换标记的状态，若切换标记的状态为未切换状态（如，VMX_OFF），将切换标记更新为切换中状态（如，VMX_SWITCH）；若切换标记的状态为切换完成状态（如，VMX_ON），则直接返回 SMP 调用，不执行模式切换。可选地，在返回 SMP 调用的基础上，可以做一些其它收尾工作。虚拟化层 30 还可以配置首次进入非根模式后的首条指令，该首条指令为切换函数的入口地址。

其中，切换函数分为上半部分和下半部分，例如，可以通过函数入口例如 switch_vcpu 进入切换函数。其中，随着宿主机操作系统在不同模式之间的切换，在不同次进入切换函数后的代码路径会有所不同。对于第一次从根模式向非根模式切换时，可以向目标物理计算资源对象发起 SMP 调用，该调用用于触发模式切换，首先检测切换标记的状态，在确定为待切换状态的情况下，触发模式切换，进入切换函数的上半部分，在上半部分先设置切换标记为切换中状态（如，VMX_SWITCH），并设置进入非根模式后执行的第一条指令，即切换函数的入口地址。此后，在切换函数的上半部分中，执行将宿主机操作系统从根模式切换为非根模式的操作，目标物理计算资源对象进入非根模式后执行上述设置的首条指令，首条指令为执行切换函数，此时，标志位已经设置为切换中状态，则进入切换函数的下半部分，在切换函数的下半部分完成切换，并返回 SMP 调用。

可选地，为了便于区分和描述，将目标物理计算资源对象当前（即在切换到非根模式之前使用的栈称为第二栈。虚拟化层 30 在将目标物理计算资源对象在根模式下的上下文信息，保存至目标物理计算资源对象对应的物理描述符结构体中时，具体用于：将上下文信息中各寄存器的值和第二栈的栈顶和栈底地址保存至物理描述符结构体中。

例如，可以将上下文信息中各寄存器的值保存至物理描述符结构体中，具体可以是将控制寄存器、段寄存器和通用寄存器保存至物理描述符结构体中的寄存器结构体中，将第二栈的栈顶和栈底地址保存至物理描述符结构体（pcpu）的主机（host）寄存器的栈相应字段，后续模式切换时会用到。

虚拟化层 30 还可以将目标物理计算资源对象当前使用的第二栈，切换为新创建的第一栈，以供虚拟化层 30 在根模式下基于第一栈继续运行。

进一步可选地，虚拟化层 30 在根据虚拟化描述符结构体、虚拟化控制结构体以及内存页表，控制目标物理计算资源对象运行时，具体用于：加载虚拟化控制结构体和虚拟化描述符结构体中的专用寄存器的值，专用寄存器是一组具有特定功能的寄存器，专用寄存器具有特殊用途，能被特定的指令或者硬件模块访问和使用。例如，PI 寄存器、调试寄存器（Debug Register，DR）寄存器、MSR 寄存器或段寄存器等。进一步，还可以保存 pcpu 的段寄存器，并加载 vcpu 中的段寄存器的值。将虚拟化控制结构体中的运行控制信息注入目标物理计算资源对象，并加载虚拟化描述符结构体中的通用寄存器的值，通用寄存器是一组可以被开发人员使用的寄存器，用于存储数据和地址等通用信息。通用寄存器可以被程序中的任意指令所访问和使用。例如，指针寄存器或变址寄存器等。其中，加载相应寄存器的值，表示将 vcpu 中存储的寄存器的值加载给具体的硬件寄存器，以使目标物理计算资源对象能够按照非根模式运行。执行模式切换指令，以控制目标物理计算资源对象进入非根模式并自首条指令开始运行，在运行过程中基于内存页表进行内存管理和访问。

进一步可选地，在将运行中的宿主机操作系统从根模式切换到非根模式之后，虚拟化层 30 还可以将第二栈恢复成保存到物理描述符结构体中的栈顶和栈底的地址，并将切换标记更新为切换完成状态（如，VMX_ON）。其中，在从根模式切换到非根模式之后，将第二栈（即旧栈）恢复成保存到物理描述符结构体中的栈顶和栈底的地址，便于目标物理计算资源对象在非根模式下继续基于第二栈运行。

在本公开实施例中，考虑借助于硬件虚拟化功能，在宿主机操作系统和硬件资源之间插入一层轻量级的虚拟化层，在虚拟化层中实现虚拟资源提供功能，由虚拟化层将正在运行的宿主机操作系统从根模式切换为非根模式，从而在非根模式下提供至少一种虚拟资源并在至少一种虚拟资源上运行至少一种应用，至少一种应用至少包括宿主机操作系统在根模式下因资源不足而无法运行的应用，在无需重新安装宿主机操作系统的情况下，可以针对运行态的宿主机操作系统灵活实现虚拟资源提供功能，解决宿主机操作系统在根模式下因未开启或缺失相关系统功能导致资源不足而无法运行的应用的问题，可以让宿主机操作系统支撑更多应用的运行，有利于扩展宿主机操作系统的能力。其中，在宿主机操作系统从根模式切换为非根模式之后，可以在非根模式下提供的虚拟资源上运行的至少一种应用包括但不限于：在非根模式下，进行内存管理、文件管理、虚拟化功能、调度优化等目标功能。

在下面实施例中，以内存管理中的内存缺页管理为例，对非根模式下的内存缺页管理过程进行详细说明。在此说明，内存缺页的一种应用场景为内存交换（swap），即在内存交换过程中可能发生内存缺页的情况，但内存缺页并不仅用于内存交换过程。为了便于体现本公开实施例在非根模式下实现的内存缺页管理的优势，以将内存缺页用于内存交换为例进行说明，并先行对内存交换进行相关介绍。

在宿主机操作系统（如，Linux 系统）中的内存交换功能是一种将空闲的硬盘空间用作内存扩展的机制。当主内存（如，RAM）中的内存页面不足时，swap 功能可以将一部分不常用的内存页面移动到硬盘上，以腾出空间供其它页面使用，这个过程称为页面换出（swap out）。当进程再次使用到被换出的这部分内容时，就会触发页面换入（swap in）。

宿主机操作系统（如，Linux 系统）的 swap 功能包括两个方面：交换分区（swap partition）和交换文件（swap file）。交换分区用于交换硬盘空间，又称为交换空间。通常是在安装宿主机操作系统时分配的，也可以通过 fdisk 命令或 mkswap 等命令手动创建交换分区。将空闲的磁盘空间划分为交换分区后，可以通过 swapon 命令将其挂载为交换分区。交换分区的优点是速度快、稳定性高，但是需要提前规划好分区大小，无法动态调整。其中，fdisk 命令用于创建、删除、修改和显示磁盘分区。mkswap 命令用于创建交换分区。swapon 命令用于在物理机上启用交换分区。

交换文件是一种在普通文件系统上创建的文件，可以作为交换空间。通过 dd、fallocate 等命令创建交换文件，然后使用 mkswap 命令将其格式化为交换文件系统，最后通过 swapon 命令挂载为交换空间。其中，dd 命令和 fallocate 命令均可以用来创建文件。交换文件的优点是灵活性高，可以动态调整大小、方便管理，但是性能稍逊于交换分区。

在宿主机操作系统（如，Linux 系统）中，交换空间的大小通常设置为物理内存的两倍或三倍。如果物理内存充足，交换空间的使用率会很低，不会对物理机性能产生明显的影响。若物理内存不足，交换空间的使用率会很高，会严重影响物理机的性能。因此，交换空间的性能对于物理机性能的稳定性也是至关重要的。

另外，宿主机操作系统（如，Linux）内核的 swap 功能是针对用户进程的，不会直接影响内核页面的换入和换出。若内存不足，内核会尝试回收一些不必要的页面，并通过 swap 功能将其交换到磁盘上，以释放内存空间。这些不必要的页面包括用户进程的页面、缓存页面、匿名内存页面等，但不包括内核页面，因为，内核页面通常是不可交换的。内核代码和数据通常是锁定在内存中的，不会被交换到磁盘上，因此不会受到 swap 功能的影响。而且 swap 功能目前只支持 4K 的小页，而目前云计算中越来越多使用大页（如，2M 或 1G）的场景，还不支持大页的 swap 功能。若正在运行的操作系统没有配置 swap 功能，在线开启只能使用文件方式，性能很低，会严重影响系统的稳定性。

虽然宿主机操作系统缺失的一些系统功能是可以通过加载功能模块来实现的，但是会具有侵入性，还受制于操作系统各种已有结构体和函数功能等的约束，而且不一定能解决所有问题，比如上文提到的内核态页面的交换等。在本公开实施例中，基于虚拟化层，从根模式切换到非根模式，在非根模式下提供虚拟内存资源，使得可实现内存交换功能，不再受限于宿主机操作系统是否开启内存交换功能；另外，本公开实施例提供的解决方案更加透明，对上层应用程序来说是无感知的，能够实现更全面的功能。

进一步，在本公开实施例中，在创建内存页表时，采用第一页面粒度、第二页面粒度、第三页面粒度以及第四页面粒度相结合的混合映射粒度的方式，不仅支持小页粒度，还支持大页粒度，可以满足不同物理页映射的需求，提高内存映射的灵活度。采用小页面粒度进行内存映射，可以减小内存页面中的内部碎片，减少内存浪费；采用大页面粒度进行内存映射，减少页表项数据，减少内存页表的层级，降低内存管理开销，提高物理计算资源对象的性能。

进一步，在本公开实施例中，在创建内存页表时，不再对宿主机的物理地址空间进行限定，允许为全量物理地址空间创建内存页表即支持全量内存映射，因此既能够针对用户态页面创建内存页表，也支持针对内核态页面创建内存页表，通过支持全量物理地址空间的内存映射，为后续基于内存映射的各种操作提供基础框架，例如在基于内存映射进行内存交换时，不仅支持对用户态页面的交换，还支持对内核态页面的交换，即支持全量内存交换。下面对切换到非根模式之后，实现内存缺页管理的过程进行简单说明：

1）上述为宿主机操作系统的模式切换准备的内存页表（如，ept 页表），其客户机物理地址（GPA）和宿主机物理地址（HPA）是一一对应的。其中，正常运行在非根模式下不会触发缺页异常。为了扩展宿主机操作系统的内存管理功能，可以通过回收空闲页面、或者压缩使用页面、或者交换内存页面到磁盘、或者扩展新的虚拟内存空间等方法，回收部分内存页面（页面粒度可以是 4K 或 2M），并清空对应的内存页表的页表项，这样，在非根模式中的后续访问可以触发缺页异常。目前，可以直接清空一批空闲页面的内存页表来触发缺页异常。

2）由于步骤 1）已经清空回收了一批内存页面，可以通过一个简单的内存管理系统，比如，利用状态位进行内存管理，以便后续缺页异常使用。

3)在进行缺页异常处理之前,可以在非根模式的退出事件中设置处理函数,例如,设置退出原因 EPT 缺页关联（EXIT_REASON_EPT_VIOLATION）的缺页处理函数（ept_handler）。

4）在缺页处理函数（ept_handler），可以从底层的内存管理系统中选取一个未使用的宿主机物理页面，并将其标记为使用。

5）将未使用的宿主机的物理页面的物理地址（HPA）和发生缺页的客户机物理地

址（GPA）建立映射关系，在对应的内存页表（如，ept 页表）中建立对应页表项，以使得宿主机操作系统在非根模式下实现内存管理，这里的内存管理包括但不限于内存交换等。

需要说明的是，上述实施例所提供方法的各步骤的执行主体均可以是同一设备，或者，该方法也由不同设备作为执行主体。比如，步骤 R11 至步骤 R12 的执行主体可以为设备 A；又比如，步骤 R11 和 R12 的执行主体可以为设备 A，步骤 R13 的执行主体可以为设备 B；等等。

另外，在上述实施例及附图中的描述的一些流程中，包含了按照特定顺序出现的多个操作，但是应该清楚了解，这些操作可以不按照其在本文中出现的顺序来执行或并行执行，操作的序号如 R11 至步骤 R12 等，仅仅是用于区分开各个不同的操作，序号本身不代表任何的执行顺序。另外，这些流程可以包括更多或更少的操作，并且这些操作可以按顺序执行或并行执行。需要说明的是，本文中的“第一”、“第二”等描述，是用于区分不同的消息、设备、模块等，不代表先后顺序，也不限定“第一”和“第二”是不同的类型。

图 4 为本公开示例性实施例提供的一种地址转换装置的结构示意图，如图 4 所示，该装置包括：构建模块 41、确定模块 42、查询模块 43 以及生成模块 44。

构建模块 41，用于预先构建与 EPT 对应的第一 RMAP 表，EPT 包括至少一个 GPA 各自对应的多级页表；第一 RMAP 表中存储有至少一个 GPA 各自对应的最后一级页表项的地址信息；

确定模块 42，用于响应第一地址转换触发事件，确定需要进行地址转换的第一 GPA，查询模块 43，用于在第一 RMAP 表中查询第一 GPA 对应的最后一级页表项的地址信息；

生成模块 44，用于若查询到第一 GPA 对应的最后一级页表项的地址信息，根据第一 GPA 对应的最后一级页表项的地址信息和第一 GPA 的地址偏移量，生成第一 GPA 对应的 HPA。

在一可选实施例中，构建模块还用于：预先构建与 EPT 对应的第二 RMAP 表，第二 RMAP 表中存储有至少一个 HPA 到对应 GPA 的地址转换信息；确定模块还用于：响应第二地址转换触发事件，确定需要进行地址转换的第一 HPA，在第二 RMAP 表中查询第一 HPA 到对应 GPA 的地址转换信息；生成模块还用于：根据第一 HPA 及其到对应 GPA 的地址转换信息和第一 HPA 的地址偏移量，生成第一 HPA 对应的 GPA。

在一可选实施例中，构建模块具体用于：将 EPT 对应的 GPA 地址空间划分为至少一个地址空间，针对至少一个地址空间分别创建基础结构体，基础结构体对应设定页表级别；针对 EPT 中任一 GPA，根据任一 GPA 的地址信息和各基础结构体对应的地址空间，从各基础结构体中确定任一 GPA 对应的第一结构体；根据任一 GPA 对应的最后一级页表的级别与设定页表级别的关系，通过第一结构体记录任一 GPA 对应的最后一级页表项的地址信息。

可选地，构建模块具体用于：若任一 GPA 对应的最后一级页表的级别与设定页表级别相同，则将任一 GPA 对应的最后一级页表项的地址信息添加到第一结构体中的信息字段中，

并设置第一结构体中的标志信息为第一值；若任一 GPA 对应的最后一级页表的级别与设定页表级别不相同，则创建与任一 GPA 对应的最后一级页表的级别适配的第二结构体，将任一 GPA 对应的最后一级页表项的地址信息添加到第二结构体中的信息字段中，并将第二结构体的地址信息添加到第一结构体中的信息字段中，设置第一结构体中的标志信息为第二值。

进一步可选地，构建模块具体用于：在任一 GPA 对应的最后一级页表的级别为设定页表级别的上一级别的情况下，判断是否已经存在第二结构体；若不存在，创建与任一 GPA 对应的最后一级页表的级别适配的新结构体，作为第二结构体；在任一 GPA 对应的最后一级页表的级别为设定页表级别的下一级别的情况下，创建与任一 GPA 对应的最后一级页表的级别适配的新结构体，作为第二结构体。

在一可选实施例中，构建模块具体用于：将 EPT 对应的 HPA 地址空间划分为至少一个地址空间，针对至少一个地址空间分别创建基础结构体，基础结构体对应设定页面粒度；针对 EPT 中的任一 GPA 对应的 HPA，根据任一 HPA 的地址信息和各基础结构体对应的地址空间，从各基础结构体中确定任一 HPA 对应的第三结构体；根据任一 HPA 对应的页面粒度和设定页面粒度的关系，通过第三结构体记录任一 HPA 到对应 GPA 的地址转换信息。

可选地，构建模块具体用于：若任一 HPA 对应的页面粒度与设定页面粒度相同，则通过第三结构体中的信息字段记录任一 HPA 到对应 GPA 的地址转换信息，并设置第三结构体中的第一标志信息为第三值；若任一 HPA 对应的页面粒度与设定页面粒度不相同，则创建与任一 HPA 的页面粒度适配的第四结构体，通过第四结构体中的信息字段记录任一 HPA 到对应 GPA 的地址转换信息,并将第四结构体的地址信息添加到第三结构体中的信息字段中，设置第三结构体中的第一标志信息为第四值。

进一步可选地，构建模块具体用于：在任一 HPA 对应多个 GPA 的情况下，创建第五结构体，将第五结构体的地址信息添加到第三结构体中的信息字段中，并将任一 HPA 到多个 GPA 的地址转换信息分别添加到第五结构体中，并在第三结构体中增设表示一对多关系的标志位；在任一 HPA 对应一个 GPA 的情况下，直接将任一 HPA 到对应 GPA 的地址转换信息添加到第三结构体中的信息字段中；

构建模块具体用于：在任一 HPA 对应多个 GPA 的情况下，则创建第六结构体，将第六结构体的地址信息添加到第四结构体中的信息字段中，并将任一 HPA 到多个 GPA 的地址转换信息分别添加到第六结构体中，在第四结构体中增设表示一对多关系的标志位；在任一 HPA 对应一个 GPA 的情况下，直接将任一 HPA 到对应 GPA 的地址转换信息添加到第四结构体中的信息字段中。

在一可选实施例中，查询模块具体用于：根据第一 GPA 的地址信息和第一 RMAP 表中包含的各基础结构体对应的地址空间,从各基础结构体中确定第一 GPA 对应的第一结构体，基础结构体对应设定页表级别；通过第一 GPA 对应的第一结构体，获取第一 GPA 对应的最后一级页表项的地址信息。

可选地，查询模块具体用于：从第一结构体中获取标志信息；若标志信息为第一值，则从第一结构体中的信息字段中获取第一 GPA 对应的最后一级页表项的地址信息；若标志信息为第二值，则从第一结构体中的信息字段中获取第二结构体的地址信息，根据第二结构体的地址信息，从第二结构体的信息字段中获取第一 GPA 对应的最后一级页表项的地址信息。

在一可选实施例中，还包括如下模块：

新增模块，用于在 EPT 中新增第二 GPA 的情况下，在第一 RMAP 表中添加第二 GPA 对应的最后一级页表项的地址信息；

删除模块，用于在 EPT 中删除第三 GPA 的情况下，在第一 RMAP 表中删除第三 GPA 对应的最后一级页表项的地址信息；

修改模块，用于在 EPT 中将第四 GPA 对应的最后一级页表的第一地址信息修改为第二地址信息的情况下，在第一 RMAP 表中将第四 GPA 对应的最后一级页表的第一地址信息修改为第二地址信息；

扩展模块，用于在 EPT 中将最后一级页表扩展为下一级页表的情况下，在第一 RMAP 表中将最后一级页表中各 GPA 各自对应的最后一级页表项的地址信息修改为相应下一级页表的地址信息；

收缩模块，用于在 EPT 中地址连续的最后一级页表收缩为上一级页表的情况下，在第一 RMAP 表中将最后一级页表中各 GPA 各自对应的最后一级页表项的地址信息修改为上一级页表项的地址信息。

可选地，添加模块具体用于：根据第二 GPA 的地址信息和第一 RMAP 表中包含的各基础结构体对应的地址空间，从各基础结构体中确定第二 GPA 对应的第一结构体，第一结构体对应设定页表级别；若确定未通过第一结构体记录第二 GPA 的对应的最后一级页表项的地址信息，则根据第二 GPA 的对应的最后一级页表的级别与设定页表级别的关系，通过第一结构体记录第二 GPA 对应的最后一级页表项的地址信息。

可选地，删除模块具体用于：根据第三 GPA 的地址信息和第一 RMAP 表中包含的各基础结构体对应的地址空间，从各基础结构体中确定第三 GPA 对应的第一结构体，第一结构体对应设定页表级别；若第三 GPA 对应的最后一级页表的级别和第一结构体的标志信息指示的页表级别适配，则根据第三 GPA 对应的最后一级页表的级别与设定页表级别的关系，通过第一结构体删除第三 GPA 对应的最后一级页表项的地址信息。

可选地，扩展模块具体用于：在 EPT 中清理最后一级页表中各页表项记录的地址信息，以及清理第一 RMAP 表中最后一级页表中各 GPA 对应的最后一级页表项的地址信息；创建最后一级页表对应的下一级页表，在最后一级页表的页表项中记录下一级页表的地址信息，并在下一级页表的页表项中记录相应 HPA 对应的物理页的基地址；在第一 RMAP 表中，根据下一级页表的级别与设定页表级别的关系，通过第一结构体记录最后一级页表中的各 GPA 对应的下一级页表项的地址信息。

在可选地，收缩模块具体用于：EPT 中清理地址连续的最后一级页表中各页表项记录的地址信息，以及清理第一 RMAP 表中最后一级页表中各 GPA 对应的最后一级页表项的地址信息；创建最后一级页表对应的上一级页表，在上一级页表的页表项中分别记录相应 HPA 对应的物理页的地址信息；在第一 RMAP 表中，根据上一级页表的级别与设定页表级别的关系，通过第一结构体记录最后一级页表中的各 GPA 对应的上一级页表项的地址信息。

在一可选实施例中，该装置的执行主体为宿主机操作系统 OS 中的虚拟化管理器；或者，该装置的执行主体为宿主机 OS 与宿主机硬件资源之间的虚拟化层，虚拟化层用于将运行中的宿主机 OS 从根模式切换到非根模式。

本公开实施例还提供一种快速映射表构建装置，如图 5 所示，该装置包括：

划分模块 51，用于将 EPT 对应的 GPA 地址空间划分为至少一个地址空间，创建模块 52，用于针对至少一个地址空间分别创建基础结构体，基础结构体对应设定页表级别，EPT 包括至少一个 GPA 各自对应的多级页表；

确定模块 53，用于针对 EPT 中任一 GPA，根据任一 GPA 的地址信息和各基础结构体对应的地址空间，从各基础结构体中确定任一 GPA 对应的第一结构体；

记录模块 54，用于根据任一 GPA 对应的最后一级页表的级别与设定页表级别的关系，通过第一结构体记录任一 GPA 对应的最后一级页表项的地址信息，以构建与 EPT 对应的第一 RMAP 表，第一 RMAP 表中存储有至少一个 GPA 各自对应的最后一级页表项的地址信息。

关于本公开实施例提供的图 4 和图 5 所示装置中各步骤的详细实施方式以及有益效果已经在前述实施例中进行了详细描述，此处将不做详细阐述说明。

图 6 为本公开示例性实施例提供的一种物理机的结构示意图，该物理机又称为地址转换设备，如图 6 所示，该设备包括：存储器 64 和处理器 65。

存储器 64，用于存储计算机程序，并可被配置为存储其它各种数据以支持在地址转换设备上的操作。这些数据的示例包括用于在地址转换设备上操作的任何应用程序或方法的指令等。

处理器 65，与存储器 64 耦合，用于执行存储器 64 中的计算机程序，以用于：预先构建与 EPT 对应的第一 RMAP 表，EPT 包括至少一个 GPA 各自对应的多级页表；第一 RMAP 表中存储有至少一个 GPA 各自对应的最后一级页表项的地址信息；响应第一地址转换触发事件，确定需要进行地址转换的第一 GPA，在第一 RMAP 表中查询第一 GPA 对应的最后一级页表项的地址信息；若查询到第一 GPA 对应的最后一级页表项的地址信息，根据第一 GPA 对应的最后一级页表项的地址信息和第一 GPA 的地址偏移量，生成第一 GPA 对应的 HPA。

在一可选实施例中，处理器 65 还用于：预先构建与 EPT 对应的第二 RMAP 表，第二 RMAP 表中存储有至少一个 HPA 到对应 GPA 的地址转换信息；响应第二地址转换触发事件，确定需要进行地址转换的第一HPA，在第二RMAP表中查询第一HPA到对应GPA的地址转换信息；根据第一 HPA 及其到对应 GPA 的地址转换信息和第一 HPA 的地址偏移量，生成第一 HPA 对应的 GPA。

在一可选实施例中，处理器 65 在预先构建与 EPT 对应的第一 RMAP 表时，具体用于：将 EPT 对应的 GPA 地址空间划分为至少一个地址空间，针对至少一个地址空间分别创建基础结构体，基础结构体对应设定页表级别；针对 EPT 中任一 GPA，根据任一 GPA 的地址信息和各基础结构体对应的地址空间，从各基础结构体中确定任一 GPA 对应的第一结构体；根据任一 GPA 对应的最后一级页表的级别与设定页表级别的关系，通过第一结构体记录任一 GPA 对应的最后一级页表项的地址信息。

可选地，处理器 65 在根据任一 GPA 对应的最后一级页表的级别与设定页表级别的关系，通过第一结构体记录任一 GPA 对应的最后一级页表项的地址信息时，具体用于：若任一 GPA 对应的最后一级页表的级别与设定页表级别相同，则将任一 GPA 对应的最后一级页表项的地址信息添加到第一结构体中的信息字段中，并设置第一结构体中的标志信息为第一值；若任一 GPA 对应的最后一级页表的级别与设定页表级别不相同，则创建与任一 GPA 对应的最后一级页表的级别适配的第二结构体，将任一 GPA 对应的最后一级页表项的地址信息添加到第二结构体中的信息字段中，并将第二结构体的地址信息添加到第一结构体中的信息字段中，设置第一结构体中的标志信息为第二值。

进一步可选地，处理器 65 在创建与任一 GPA 对应的最后一级页表的级别适配的第二结构体时，具体用于：在任一 GPA 对应的最后一级页表的级别为设定页表级别的上一级别的情况下，判断是否已经存在第二结构体；若不存在，创建与任一 GPA 对应的最后一级页表的级别适配的新结构体，作为第二结构体；在任一 GPA 对应的最后一级页表的级别为设定页表级别的下一级别的情况下，创建与任一 GPA 对应的最后一级页表的级别适配的新结构体，作为第二结构体。

在一可选实施例中，处理器 65 在预先构建与 EPT 对应的第二 RMAP 表时，具体用于：将 EPT 对应的 HPA 地址空间划分为至少一个地址空间，针对至少一个地址空间分别创建基础结构体，基础结构体对应设定页面粒度；针对 EPT 中的任一 GPA 对应的 HPA，根据任一 HPA 的地址信息和各基础结构体对应的地址空间，从各基础结构体中确定任一 HPA 对应的第三结构体；根据任一 HPA 对应的页面粒度和设定页面粒度的关系，通过第三结构体记录任一 HPA 到对应 GPA 的地址转换信息。

可选地，处理器 65 在根据任一 HPA 中对应的页表级别和设定页表级别的关系，通过第三结构体记录任一 HPA 到对应 GPA 的地址转换信息时，具体用于：若任一 HPA 对应的页面粒度与设定页面粒度相同，则通过第三结构体中的信息字段记录任一 HPA 到对应 GPA 的地址转换信息，并设置第三结构体中的第一标志信息为第三值；若任一 HPA 对应的页面粒度与设定页面粒度不相同，则创建与任一 HPA 的页面粒度适配的第四结构体，通过第四结构体中的信息字段记录任一 HPA 到对应 GPA 的地址转换信息，并将第四结构体的地址信息添加到第三结构体中的信息字段中，设置第三结构体中的第一标志信息为第四值。

进一步可选地，处理器 65 在通过第三结构体中的信息字段记录任一 HPA 到对应 GPA 的地址转换信息时，具体用于：在任一 HPA 对应多个 GPA 的情况下，创建第五结构体，将

第五结构体的地址信息添加到第三结构体中的信息字段中，并将任一 HPA 到多个 GPA 的地址转换信息分别添加到第五结构体中，并在第三结构体中增设表示一对多关系的标志位；在任一 HPA 对应一个 GPA 的情况下，直接将任一 HPA 到对应 GPA 的地址转换信息添加到第三结构体中的信息字段中；

相应地，处理器 65 在通过第四结构体中的信息字段记录任一 HPA 到对应 GPA 的地址转换信息时，具体用于：在任一 HPA 对应多个 GPA 的情况下，则创建第六结构体，将第六结构体的地址信息添加到第四结构体中的信息字段中，并将任一 HPA 到多个 GPA 的地址转换信息分别添加到第六结构体中，在第四结构体中增设表示一对多关系的标志位；在任一 HPA 对应一个 GPA 的情况下，直接将任一 HPA 到对应 GPA 的地址转换信息添加到第四结构体中的信息字段中。

在一可选实施例中，处理器 65 在第一 RMAP 表中查询第一 GPA 对应的最后一级页表项的地址信息时，具体用于：根据第一 GPA 的地址信息和第一 RMAP 表中包含的各基础结构体对应的地址空间，从各基础结构体中确定第一 GPA 对应的第一结构体，基础结构体对应设定页表级别；通过第一 GPA 对应的第一结构体，获取第一 GPA 对应的最后一级页表项的地址信息。

可选地，处理器 65 在通过第一 GPA 对应的第一结构体，获取第一 GPA 对应的最后一级页表项的地址信息时，具体用于：从第一结构体中获取标志信息；若标志信息为第一值，则从第一结构体中的信息字段中获取第一 GPA 对应的最后一级页表项的地址信息；若标志信息为第二值，则从第一结构体中的信息字段中获取第二结构体的地址信息，根据第二结构体的地址信息，从第二结构体的信息字段中获取第一 GPA 对应的最后一级页表项的地址信息。

在一可选实施例中，处理器 65 还用于执行以下至少一种操作：

在 EPT 中新增第二 GPA 的情况下，在第一 RMAP 表中添加第二 GPA 对应的最后一级页表项的地址信息；

在 EPT 中删除第三 GPA 的情况下，在第一 RMAP 表中删除第三 GPA 对应的最后一级页表项的地址信息；

在 EPT 中将第四 GPA 对应的最后一级页表的第一地址信息修改为第二地址信息的情况下，在第一 RMAP 表中将第四 GPA 对应的最后一级页表的第一地址信息修改为第二地址信息；

在 EPT 中将最后一级页表扩展为下一级页表的情况下，在第一 RMAP 表中将最后一级页表中各 GPA 各自对应的最后一级页表项的地址信息修改为相应下一级页表的地址信息；

在 EPT 中地址连续的最后一级页表收缩为上一级页表的情况下，在第一 RMAP 表中将最后一级页表中各 GPA 各自对应的最后一级页表项的地址信息修改为上一级页表项的地址信息。

在一可选实施例中，处理器 65 在在第一 RMAP 表中添加第二 GPA 对应的最后一级页表

项的地址信息时，具体用于：根据第二 GPA 的地址信息和第一 RMAP 表中包含的各基础结构体对应的地址空间，从各基础结构体中确定第二 GPA 对应的第一结构体，第一结构体对应设定页表级别；若确定未通过第一结构体记录第二 GPA 的对应的最后一级页表项的地址信息，则根据第二 GPA 的对应的最后一级页表的级别与设定页表级别的关系，通过第一结构体记录第二 GPA 对应的最后一级页表项的地址信息。

在一可选实施例中，处理器 65 在在第一 RMAP 表中删除第三 GPA 对应的最后一级页表项的地址信息时，具体用于：根据第三 GPA 的地址信息和第一 RMAP 表中包含的各基础结构体对应的地址空间，从各基础结构体中确定第三 GPA 对应的第一结构体，第一结构体对应设定页表级别；若第三 GPA 对应的最后一级页表的级别和第一结构体的标志信息指示的页表级别适配，则根据第三 GPA 对应的最后一级页表的级别与设定页表级别的关系，通过第一结构体删除第三 GPA 对应的最后一级页表项的地址信息。

在一可选实施例中，处理器 65 在在 EPT 中将最后一级页表扩展为下一级页表的情况下，在第一 RMAP 表中将最后一级页表中各 GPA 各自对应的最后一级页表项的地址信息修改为相应下一级页表的地址信息时，具体用于：在 EPT 中清理最后一级页表中各页表项记录的地址信息，以及清理第一 RMAP 表中最后一级页表中各 GPA 对应的最后一级页表项的地址信息；创建最后一级页表对应的下一级页表，在最后一级页表的页表项中记录下一级页表的地址信息，并在下一级页表的页表项中记录相应 HPA 对应的物理页的基地址；在第一 RMAP 表中，根据下一级页表的级别与设定页表级别的关系，通过第一结构体记录最后一级页表中的各 GPA 对应的下一级页表项的地址信息。

在一可选实施例中，处理器 65 在在 EPT 中地址连续的最后一级页表收缩为上一级页表的情况下，在第一 RMAP 表中将最后一级页表中各 GPA 各自对应的最后一级页表项的地址信息修改为上一级页表项的地址信息时，具体用于：在 EPT 中清理地址连续的最后一级页表中各页表项记录的地址信息，以及清理第一 RMAP 表中最后一级页表中各 GPA 对应的最后一级页表项的地址信息；创建最后一级页表对应的上一级页表，在上一级页表的页表项中分别记录相应 HPA 对应的物理页的地址信息；在第一 RMAP 表中，根据上一级页表的级别与设定页表级别的关系，通过第一结构体记录最后一级页表中的各 GPA 对应的上一级页表项的地址信息。

在一可选实施例中，处理器 65 通过宿主机操作系统 OS 中的虚拟化管理器执行上述操作；或者，处理器 65 通过宿主机 OS 与宿主机硬件资源之间的虚拟化层执行上述操作，虚拟化层用于将运行中的宿主机 OS 从根模式切换到非根模式。

关于本公开实施例提供的图 6 所示设备中各步骤的详细实施方式以及有益效果已经在前述实施例中进行了详细描述，此处将不做详细阐述说明。

进一步，如图 6 所示，该地址转换设备还包括：通信组件 66、显示器 67、电源组件 68、音频组件 69 等其它组件。图 6 中仅示意性给出部分组件，并不意味着地址转换设备只包括图 6 所示组件。另外，图 6 中虚线框内的组件为可选组件，而非必选组

件，具体可视地址转换设备的产品形态而定。本实施例的地址转换设备可以实现为台式电脑、笔记本电脑、智能手机或 IOT 设备等终端设备，也可以是常规服务器、云服务器或服务器阵列等服务端设备。若本实施例的地址转换设备实现为台式电脑、笔记本电脑、智能手机等终端设备，可以包含图 6 中虚线框内的组件；若本实施例的地址转换设备实现为常规服务器、云服务器或服务器阵列等服务端设备，则可以不包含图 6 中虚线框内的组件。

本公开实施例还提供一种快速映射的构建设备，该快速映射的构建设备的实现结构与图 6 所示快速映射的构建设备的实现结构相同或类似，可参照图 6 所示快速映射的构建设备的结构实现。本实施例提供的快速映射的构建设备与图 6 所示实施例中快速映射的构建设备的区别主要在于：处理器执行存储器中存储的计算机程序所实现的功能不同。对本实施例提供的地址转换设备来说，其处理器执行存储器中存储的计算机程序，可用于：将内存页表 EPT 对应的 GPA 地址空间划分为至少一个地址空间，针对至少一个地址空间分别创建基础结构体，基础结构体对应设定页表级别，EPT 包括至少一个 GPA 各自对应的多级页表；针对 EPT 中任一 GPA，根据任一 GPA 的地址信息和各基础结构体对应的地址空间，从各基础结构体中确定任一 GPA 对应的第一结构体；根据任一 GPA 对应的最后一级页表的级别与设定页表级别的关系，通过第一结构体记录任一 GPA 对应的最后一级页表项的地址信息，以构建与 EPT 对应的第一 RMAP 表，第一 RMAP 表中存储有至少一个 GPA 各自对应的最后一级页表项的地址信息。

关于本公开实施例提供的快速映射的构建设备的详细实施方式以及有益效果已经在前述实施例中进行了详细描述，此处将不做详细阐述说明。

相应地，本公开实施例还提供一种存储有计算机程序的计算机可读存储介质，计算机程序被执行时能够实现上述图 4 和图 5 所示方法实施例中的各步骤。

本公开实施例还提供一种计算机程序产品，包括计算机程序/指令，当计算机程序/指令被处理器执行时，致使处理器实现述图 4 和图 5 所示方法中的步骤。

上述存储器可以由任何类型的易失性或非易失性存储设备或者它们的组合实现，如静态随机存取存储器（Static Random-Access Memory，SRAM），电可擦除可编程只读存储器（Electrically Erasable Programmable Read Only Memory，EEPROM），可擦除可编程只读存储器（Erasable Programmable Read Only Memory，EPROM），可编程只读存储器（Programmable Read-Only Memory，PROM），只读存储器（Read-Only Memory，ROM），磁存储器，快闪存储器，磁盘或光盘。

上述通信组件被配置为便于通信组件所在设备和其他设备之间有线或无线方式的通信。通信组件所在设备可以接入基于通信标准的无线网络，如 WiFi，2G、3G、4G/LTE、5G 等移动通信网络，或它们的组合。在一个示例性实施例中，通信组件经由广播信道接收来自外部广播管理系统的广播信号或广播相关信息。在一个示例性实施例中，通

信组件还包括近场通信（Near Field Communication，NFC）模块，以促进短程通信。例如，在 NFC 模块可基于射频识别（Radio Frequency Identification，RFID）技术，红外数据协会（Infrared Data Association，IrDA）技术，超宽带（Ultra Wide Band，UWB）技术，蓝牙（BlueTooth，BT）技术和其他技术来实现。

上述显示器包括屏幕，其屏幕可以包括液晶显示器（Liquid Crystal Display，LCD）和触摸面板（TouchPanel，TP）。如果屏幕包括触摸面板，屏幕可以被实现为触摸屏，以接收来自用户的输入信号。触摸面板包括一个或多个触摸传感器以感测触摸、滑动和触摸面板上的手势。触摸传感器可以不仅感测触摸或滑动动作的边界，而且还检测与触摸或滑动操作相关的持续时间和压力。

上述电源组件，为电源组件所在设备的各种组件提供电力。电源组件可以包括电源管理系统，一个或多个电源，及其他与为电源组件所在设备生成、管理和分配电力相关联的组件。

上述音频组件，可被配置为输出和/或输入音频信号。例如，音频组件包括一个麦克风（Microphone，MIC），当音频组件所在设备处于操作模式，如呼叫模式、记录模式和语音识别模式时，麦克风被配置为接收外部音频信号。所接收的音频信号可以被进一步存储在存储器或经由通信组件发送。在一些实施例中，音频组件还包括一个扬声器，用于输出音频信号。

本领域内的技术人员应明白，本公开的实施例可提供为方法、系统、或计算机程序产品。因此，本公开可采用完全硬件实施例、完全软件实施例、或结合软件和硬件方面的实施例的形式。而且，本公开可采用在一个或多个其中包含有计算机可用程序代码的计算机可读存储介质（包括但不限于磁盘存储器、只读光盘（Compact Disc Read-Only Memory，CD-ROM）、光学存储器等）上实施的计算机程序产品的形式。

本公开是参照根据本公开实施例的方法、设备（系统）、和计算机程序产品的流程图和/或方框图来描述的。应理解可由计算机程序指令实现流程图和/或方框图中的每一流程和/或方框、以及流程图和/或方框图中的流程和/或方框的结合。可提供这些计算机程序指令到通用计算机、专用计算机、嵌入式处理机或其他可编程数据处理设备的处理器以产生一个机器，使得通过计算机或其他可编程数据处理设备的处理器执行的指令产生用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的装置。

这些计算机程序指令也可存储在能引导计算机或其他可编程数据处理设备以特定方式工作的计算机可读存储器中，使得存储在该计算机可读存储器中的指令产生包括指令装置的制造品，该指令装置实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能。

这些计算机程序指令也可装载到计算机或其他可编程数据处理设备上，使得在计算机或其他可编程设备上执行一系列操作步骤以产生计算机实现的处理，从而在计算

机或其他可编程设备上执行的指令提供用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的步骤。

在一个典型的配置中，计算设备包括一个或多个处理器（Central Processing Unit，CPU）、输入/输出接口、网络接口和内存。

内存可能包括计算机可读介质中的非永久性存储器，随机存取存储器（Random Access Memory，RAM）和/或非易失性内存等形式，如只读存储器（ROM）或闪存（flash RAM）。内存是计算机可读介质的示例。

计算机可读介质包括永久性和非永久性、可移动和非可移动媒体可以由任何方法或技术来实现信息存储。信息可以是计算机可读指令、数据结构、程序的模块或其他数据。计算机的存储介质的例子包括，但不限于相变内存（Phase-change Random Access Memory，PRAM）、静态随机存取存储器（SRAM）、动态随机存取存储器（Dynamic Random Access Memory，DRAM）、其他类型的随机存取存储器（RAM）、只读存储器（ROM）、电可擦除可编程只读存储器（EEPROM）、快闪记忆体或其他内存技术、只读光盘只读存储器（CD-ROM）、数字多功能光盘（Digital Video Disc，DVD）或其他光学存储、磁盒式磁带，磁带磁盘存储或其他磁性存储设备或任何其他非传输介质，可用于存储可以被计算设备访问的信息。按照本文中的界定，计算机可读介质不包括暂存电脑可读媒体（transitory media），如调制的数据信号和载波。

还需要说明的是，术语“包括”、“包含”或者其任何其他变体意在涵盖非排他性的包含，从而使得包括一系列要素的过程、方法、商品或者设备不仅包括那些要素，而且还包括没有明确列出的其他要素，或者是还包括为这种过程、方法、商品或者设备所固有的要素。在没有更多限制的情况下，由语句“包括一个……”限定的要素，并不排除在包括要素的过程、方法、商品或者设备中还存在另外的相同要素。

以上仅为本公开的实施例而已，并不用于限制本公开。对于本领域技术人员来说，本公开可以有各种更改和变化。凡在本公开的精神和原理之内所作的任何修改、等同替换、改进等，均应包含在本公开的权利要求范围之内。

权 利 要 求 书

1、一种地址转换方法，其中，包括：

预先构建与内存页表对应的第一快速映射表，所述内存页表包括至少一个客户机物理地址各自对应的多级页表；所述第一快速映射表中存储有所述至少一个客户机物理地址各自对应的最后一级页表项的地址信息；

响应第一地址转换触发事件，确定需要进行地址转换的第一客户机物理地址，在所述第一快速映射表中查询所述第一客户机物理地址对应的最后一级页表项的地址信息；

若查询到所述第一客户机物理地址对应的最后一级页表项的地址信息，根据所述第一客户机物理地址对应的最后一级页表项的地址信息和所述第一客户机物理地址的地址偏移量，生成所述第一客户机物理地址对应的宿主机物理地址。

2、根据权利要求 1 所述的方法，其中，还包括：

预先构建与内存页表对应的第二快速映射表，所述第二快速映射表中存储有至少一个宿主机物理地址到对应客户机物理地址的地址转换信息；

响应第二地址转换触发事件，确定需要进行地址转换的第一宿主机物理地址，在所述第二快速映射表中查询所述第一宿主机物理地址到对应客户机物理地址的地址转换信息；

根据所述第一宿主机物理地址及其到对应客户机物理地址的地址转换信息和所述第一宿主机物理地址的地址偏移量，生成所述第一宿主机物理地址对应的客户机物理地址。

3、根据权利要求 1 或 2 所述的方法，其中，预先构建与内存页表对应的第一快速映射表，包括：

将所述内存页表对应的客户机物理地址的地址空间划分为至少一个地址空间，针对所述至少一个地址空间分别创建基础结构体，所述基础结构体对应设定页表级别；

针对所述内存页表中任一客户机物理地址，根据所述任一客户机物理地址的地址信息和各基础结构体对应的地址空间，从各基础结构体中确定所述任一客户机物理地址对应的第一结构体；

根据所述任一客户机物理地址对应的最后一级页表的级别与设定页表级别的关系，通过所述第一结构体记录所述任一客户机物理地址对应的最后一级页表项的地址信息。

4、根据权利要求 3 所述的方法，其中，根据所述任一客户机物理地址对应的最后一级页表的级别与设定页表级别的关系，通过所述第一结构体记录所述任一客户机物理地址对应的最后一级页表项的地址信息，包括：

若所述任一客户机物理地址对应的最后一级页表的级别与设定页表级别相同，则将所述任一客户机物理地址对应的最后一级页表项的地址信息添加到所述第一结构体中的信息字段中，并设置所述第一结构体中的标志信息为第一值；

若所述任一客户机物理地址对应的最后一级页表的级别与设定页表级别不相同，则创建与所述任一客户机物理地址对应的最后一级页表的级别适配的第二结构体，将所述任一客户机物理地址对应的最后一级页表项的地址信息添加到所述第二结构体中的信息字段

中，并将所述第二结构体的地址信息添加到所述第一结构体中的信息字段中，设置所述第一结构体中的标志信息为第二值。

5、根据权利要求 4 所述的方法，其中，创建与所述任一客户机物理地址对应的最后一级页表的级别适配的第二结构体，包括：

在所述任一客户机物理地址对应的最后一级页表的级别为设定页表级别的上一级别的情况下，判断是否已经存在所述第二结构体；若不存在，创建与所述任一客户机物理地址对应的最后一级页表的级别适配的新结构体，作为所述第二结构体；

在所述任一客户机物理地址对应的最后一级页表的级别为设定页表级别的下一级别的情况下，创建与所述任一客户机物理地址对应的最后一级页表的级别适配的新结构体，作为所述第二结构体。

6、根据权利要求 2 所述的方法，其中，预先构建与内存页表对应的第二快速映射表，包括：

将所述内存页表对应的宿主机物理地址的地址空间划分为至少一个地址空间，针对所述至少一个地址空间分别创建基础结构体，所述基础结构体对应设定页面粒度；

针对所述内存页表中的任一客户机物理地址对应的宿主机物理地址，根据任一宿主机物理地址的地址信息和各基础结构体对应的地址空间，从各基础结构体中确定所述任一宿主机物理地址对应的第三结构体；

根据所述任一宿主机物理地址对应的页面粒度和设定页面粒度的关系，通过所述第三结构体记录所述任一宿主机物理地址到对应客户机物理地址的地址转换信息。

7、根据权利要求 6 所述的方法，其中，根据所述任一宿主机物理地址中对应的页表级别和设定页表级别的关系，通过所述第三结构体记录所述任一宿主机物理地址到对应客户机物理地址的地址转换信息，包括：

若所述任一宿主机物理地址对应的页面粒度与设定页面粒度相同，则通过所述第三结构体中的信息字段记录所述任一宿主机物理地址到对应客户机物理地址的地址转换信息，并设置所述第三结构体中的第一标志信息为第三值；

若所述任一宿主机物理地址对应的页面粒度与设定页面粒度不相同，则创建与所述任一宿主机物理地址的页面粒度适配的第四结构体，通过所述第四结构体中的信息字段记录所述任一宿主机物理地址到对应客户机物理地址的地址转换信息，并将所述第四结构体的地址信息添加到所述第三结构体中的信息字段中，设置所述第三结构体中的第一标志信息为第四值。

8、根据权利要求 7 所述的方法，其中，通过所述第三结构体中的信息字段记录所述任一宿主机物理地址到对应客户机物理地址的地址转换信息，包括：

在所述任一宿主机物理地址对应多个客户机物理地址的情况下，创建第五结构体，将所述第五结构体的地址信息添加到所述第三结构体中的信息字段中，并将所述任一宿主机物理地址到多个客户机物理地址的地址转换信息分别添加到所述第五结构体中，并在所述

第三结构体中增设表示一对多关系的标志位；

在所述任一宿主机物理地址对应一个客户机物理地址的情况下，直接将所述任一宿主机物理地址到对应客户机物理地址的地址转换信息添加到所述第三结构体中的信息字段中；

相应地，通过所述第四结构体中的信息字段记录所述任一宿主机物理地址到对应客户机物理地址的地址转换信息，包括：

在所述任一宿主机物理地址对应多个客户机物理地址的情况下，则创建第六结构体，将所述第六结构体的地址信息添加到所述第四结构体中的信息字段中，并将所述任一宿主机物理地址到多个客户机物理地址的地址转换信息分别添加到所述第六结构体中，在所述第四结构体中增设表示一对多关系的标志位；

在所述任一宿主机物理地址对应一个客户机物理地址的情况下，直接将所述任一宿主机物理地址到对应客户机物理地址的地址转换信息添加到所述第四结构体中的信息字段中。

9、根据权利要求 1-8 任一项所述的方法，其中，在所述第一快速映射表中查询所述第一客户机物理地址对应的最后一级页表项的地址信息，包括：

根据所述第一客户机物理地址的地址信息和所述第一快速映射表中包含的各基础结构体对应的地址空间，从各基础结构体中确定所述第一客户机物理地址对应的第一结构体，所述基础结构体对应设定页表级别；

通过所述第一客户机物理地址对应的第一结构体，获取所述第一客户机物理地址对应的最后一级页表项的地址信息。

10、根据权利要求 9 所述的方法，其中，通过所述第一客户机物理地址对应的第一结构体，获取所述第一客户机物理地址对应的最后一级页表项的地址信息，包括：

从所述第一结构体中获取标志信息；

若所述标志信息为第一值，则从所述第一结构体中的信息字段中获取所述第一客户机物理地址对应的最后一级页表项的地址信息；

若所述标志信息为第二值，则从所述第一结构体中的信息字段中获取第二结构体的地址信息，根据所述第二结构体的地址信息，从所述第二结构体的信息字段中获取所述第一客户机物理地址对应的最后一级页表项的地址信息。

11、根据权利要求 1-2，4-10 任一项所述的方法，其中，还包括以下至少一种操作：

在所述内存页表中新增第二客户机物理地址的情况下，在所述第一快速映射表中添加所述第二客户机物理地址对应的最后一级页表项的地址信息；

在所述内存页表中删除第三客户机物理地址的情况下，在所述第一快速映射表中删除所述第三客户机物理地址对应的最后一级页表项的地址信息；

在所述内存页表中将第四客户机物理地址对应的最后一级页表的第一地址信息修改为第二地址信息的情况下，在所述第一快速映射表中将所述第四客户机物理地址对应的最

后一级页表的第一地址信息修改为第二地址信息；

在所述内存页表中将最后一级页表扩展为下一级页表的情况下，在所述第一快速映射表中将所述最后一级页表中各客户机物理地址各自对应的最后一级页表项的地址信息修改为所述下一级页表的地址信息；

在所述内存页表中地址连续的最后一级页表收缩为上一级页表的情况下，在所述第一快速映射表中清理所述最后一级页表项的地址信息，并添加所述最后一级页表中各客户机物理地址对应的所述上一级页表项的地址信息。

12、根据权利要求 11 所述的方法，其中，在所述第一快速映射表中添加所述第二客户机物理地址对应的最后一级页表项的地址信息，包括：

根据所述第二客户机物理地址的地址信息和所述第一快速映射表中包含的各基础结构体对应的地址空间，从各基础结构体中确定所述第二客户机物理地址对应的第一结构体，所述第一结构体对应设定页表级别；

若确定未通过所述第一结构体记录所述第二客户机物理地址的对应的最后一级页表项的地址信息，则根据所述第二客户机物理地址的对应的最后一级页表的级别与设定页表级别的关系，通过所述第一结构体记录所述第二客户机物理地址对应的最后一级页表项的地址信息。

13、根据权利要求 11 所述的方法，其中，在所述第一快速映射表中删除所述第三客户机物理地址对应的最后一级页表项的地址信息，包括：

根据所述第三客户机物理地址的地址信息和所述第一快速映射表中包含的各基础结构体对应的地址空间，从各基础结构体中确定所述第三客户机物理地址对应的第一结构体，所述第一结构体对应设定页表级别；

若所述第三客户机物理地址对应的最后一级页表的级别和所述第一结构体的标志信息指示的页表级别适配，则根据所述第三客户机物理地址对应的最后一级页表的级别与设定页表级别的关系，通过所述第一结构体删除所述第三客户机物理地址对应的最后一级页表项的地址信息。

14、根据权利要求 11 所述的方法，其中，在所述内存页表中将最后一级页表扩展为下一级页表的情况下，在所述第一快速映射表中将所述最后一级页表中各客户机物理地址各自对应的最后一级页表项的地址信息修改为所述下一级页表的地址信息，包括：

在所述内存页表中清理所述最后一级页表中各页表项记录的地址信息，以及清理所述第一快速映射表中所述最后一级页表中各客户机物理地址对应的最后一级页表项的地址信息；

创建所述最后一级页表对应的下一级页表，在所述最后一级页表的页表项中记录所述下一级页表的地址信息，并在所述下一级页表的页表项中记录相应宿主机物理地址对应的物理页的基地址；

在第一快速映射表中，根据所述下一级页表的级别与设定页表级别的关系，通过第一

结构体记录所述最后一级页表中的各客户机物理地址对应的下一级页表项的地址信息。

15、根据权利要求 11 所述的方法，其中，在所述内存页表中地址连续的最后一级页表收缩为上一级页表的情况下，在所述第一快速映射表中清理所述最后一级页表项的地址信息，并添加所述最后一级页表中各客户机物理地址对应的所述上一级页表项的地址信息，包括：

在所述内存页表中清理所述地址连续的最后一级页表中各页表项记录的地址信息，以及清理所述第一快速映射表中所述最后一级页表中各客户机物理地址对应的最后一级页表项的地址信息；

在内存页表中创建所述最后一级页表对应的上一级页表，按照所述上一级页表项对应的页面粒度，在上一级页表的页表项中分别记录相应宿主机物理地址对应的物理页的基地址；

在第一快速映射表中，清理所述最后一级页表项的地址信息，根据所述上一级页表的级别与设定页表级别的关系，通过第一结构体记录所述最后一级页表中的各客户机物理地址对应的上一级页表项的地址信息。

16、根据权利要求 1-2，4-10，12-15 任一项所述的方法，其中，所述方法的执行主体为宿主机操作系统 OS 中的虚拟化管理器；或者，所述方法的执行主体为宿主机 OS 与宿主机硬件资源之间的虚拟化层，所述虚拟化层用于将运行中的宿主机 OS 从根模式切换到非根模式。

17、一种快速映射表构建方法，其中，包括：

将内存页表对应的客户机物理地址的地址空间划分为至少一个地址空间，针对所述至少一个地址空间分别创建基础结构体，所述基础结构体对应设定页表级别，所述内存页表包括至少一个客户机物理地址各自对应的多级页表；

针对所述内存页表中任一客户机物理地址，根据所述任一客户机物理地址的地址信息和各基础结构体对应的地址空间，从各基础结构体中确定所述任一客户机物理地址对应的第一结构体；

根据所述任一客户机物理地址对应的最后一级页表的级别与设定页表级别的关系，通过所述第一结构体记录所述任一客户机物理地址对应的最后一级页表项的地址信息，以构建与所述内存页表对应的第一快速映射表，所述第一快速映射表中存储有所述至少一个客户机物理地址各自对应的最后一级页表项的地址信息。

18、一种物理机，其中，包括：存储器和处理器；所述存储器，用于存储计算机程序；所述处理器，与所述存储器耦合，用于执行所述计算机程序，以实现权利要求 1-16 以及权利要求 17 中任一项所述方法中的步骤。

19、一种存储有计算机程序的计算机可读存储介质，其中，当所述计算机程序被处理器执行时，致使所述处理器实现权利要求 1-16 以及权利要求 17 中任一项所述方法中的步骤。

20、一种计算机程序产品，包括计算机程序/指令，其中，当所述计算机程序/指令被处理器执行时，致使所述处理器实现权利要求 1-16 和权利要求 17 中的任一项所述方法中的步骤。