

说明书

一种显存的分配方法、请求方法、设备及存储介质

本公开要求于 2023 年 09 月 01 日提交中国专利局的申请号为 202311125060.1、申请名称为“一种显存的分配方法、请求方法、设备及存储介质”的优先权，其全部内容
5 通过引用结合在本公开中。

技术领域

本公开涉及异构计算技术领域，尤其涉及一种显存的分配方法、请求方法、设备及存储介质。
10

背景技术

云渲染是异构计算技术领域中的一个重要场景。广义上的云渲染包括了云桌面，云手机，云游戏，元宇宙等场景。它的目的是将传统的需要本地渲染的应用，放到性能强大的云端异构服务器上去执行。例如：云游戏就是将传统需要本地渲染的游戏程序，放到云端异构服务器上，利用云端异构服务器的 GPU 进行渲染，然后将得到的结果通过网络取回，最终在本地进行显示。
15

目前，考虑到不同应用通常不会同时占用 GPU 所关联的显存，因此，一个 GPU 可同时为多个应用提供云渲染服务。但是，应用为了提高其自身运行的流畅度，通常会占用尽量多的显存，而 GPU 所关联的显存是有限的，这就可能导致 GPU 所服务的其它应用分配不到足量的显存，进而导致应用退出或者性能下降，最终导致用户体验的下降。
20

发明内容

本公开的多个方面提供一种显存的分配方法、请求方法、设备及存储介质，用以
25 保证共用显存的各个应用程序的性能。

本公开实施例提供一种显存的分配方法，适用于物理机，所述物理机上装配有显存，所述方法包括：

在拦截到所述物理机上运行的任一应用程序发起的显存占用请求的情况下，确定所述应用程序所请求的待占用显存量；

30 判断所述待占用显存量和所述应用程序已占用的显存量之和是否超过所述应用程序对应的显存占用上限值；

若超过，则拒绝响应所述显存占用请求。

本公开实施例还提供一种显存的请求方法，适用于物理机上的应用程序，所述物理机上装配有显存，所述方法包括：

向所述物理机上的应用渲染组件发起显存占用请求，所述显存占用请求中携带有待占用显存量；

若接收到所述应用渲染组件反馈的响应失败通知，则向所述应用渲染组件发起重试请求，所述重试请求中携带的待占用显存量低于响应失败的所述显存占用请求中所携带的待占用显存量；

其中，所述响应失败通知是由所述应用渲染组件在接收到所述物理机上增设的显存过滤组件发送的拒绝指令后反馈至所述应用程序的，所述显存过滤组件用于对所述应用渲染组件中的显存占用请求进行拦截并在确定显存占用请求不符合放行条件的情况下向应用渲染组件发送拒绝指令。

本公开实施例还提供一种物理机，包括存储器和处理器，所述物理机上装配有显存；

所述存储器用于存储一条或多条计算机指令；

所述处理器与所述存储器和所述显存耦合，用于执行所述一条或多条计算机指令，以用于执行前述的显存的分配方法或者显存的请求方法。

本公开实施例还提供一种存储计算机指令的计算机可读存储介质，当所述计算机指令被一个或多个处理器执行时，致使所述一个或多个处理器执行前述的显存的分配方法或者显存的请求方法。

在本公开实施例中，提出对物理机中应用程序发起的显存占用请求进行拦截，并为物理机中的各个应用程序分别配置显存占用上限值；基于此，在拦截到显存占用请求的情况下，可判断应用程序本次所请求的待占用显存量和应用程序已占用的显存量之和是否已经超过了应用程序对应的显存占用上限值，若已经超过，则拒绝响应显存占用请求。这样，可基于为不同应用程序配置的显存占用上限值，对物理机中发生的显存占用请求进行过滤，以实现针对不同应用程序所占用的显存进行有效隔离，从而避免显存被部分应用程序过量占用的问题，进而保证共用显存的各个应用程序的正常性能。

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 1 为本公开一示例性实施例提供的一种显存的分配方法的流程示意图；

图 2 为本公开一示例性实施例提供的一种显存的分配方法的逻辑示意图；

图 3 为本公开一示例性实施例提供的一种显存的分配方法的可选实现方式的逻辑示意图；

图 4 为本公开一示例性实施例提供的另一种显存的分配方法的流程示意图；

图 5 为本公开另一示例性实施例提供的一种显存的请求方法的流程示意图；
图 6 为本公开一示例性实施例提供的一种应用场景的示意图；
图 7 为本公开又一示例性实施例提供的一种物理机的结构示意图。

5 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

10 围。

目前，应用为了提高其自身运行的流畅度，通常会占用尽量多的显存，这就可能导致 GPU 所服务的其它应用分配不到足量的显存，进而导致应用退出或者性能下降，最终导致用户体验的下降。为此，本公开的一些实施例中：提出对物理机中应用程序发起的显存占用请求进行拦截，并为物理机中的各个应用程序分别配置显存占用上限值；基于此，在拦截到显存占用请求的情况下，可判断应用程序本次所请求的待占用显存量和应用程序已占用的显存量之和是否已经超过了应用程序对应的显存占用上限值，若已经超过，则拒绝响应显存占用请求。这样，可基于为不同应用程序配置的显存占用上限值，对物理机中发生的显存占用请求进行过滤，以实现对不同应用程序所占用的显存进行有效隔离，从而避免显存被部分应用程序过量占用的问题，进而保证

15 共用显存的各个应用程序的正常性能。

20 以下结合附图，详细说明本公开各实施例提供的技术方案。

图 1 为本公开一示例性实施例提供的一种显存的分配方法的流程示意图，图 2 为本公开一示例性实施例提供的一种显存的分配方法的逻辑示意图。该方法可由显存过滤组件执行，该显存过滤组件可实现为软件、硬件或软件与硬件的结合，该显存过滤组件可集成在物理机中。参考图 2，物理机上装配有显存。显存作为显卡的重要组成部分，可接入物理机中的总线，以基于总线协议与物理机中的处理器等部件耦合。

25 其中，显存，可理解为显卡的内存，也可被称为帧缓存，它的作用是用来存储显示芯片处理过或者即将提取的渲染数据。本实施例中，不限定显卡芯片的类型及规格等，本实施例中的显卡芯片可以是指 GPU（graphics processing unit，图像处理单元）。

值得说明的是，本实施例中的 GPU 可作为物理机的异构加速器件，GPU 可为物理机上运行的各个应用程序提供加速服务。

30 本实施例提供的显存的分配方法可应用于各种使用 GPU 进行异构加速的场景中。在云桌面场景、云游戏场景或云手机场景等云渲染场景下，本实施例中的物理机可以是云服务器。其中，云渲染类应用程序可理解为在云服务器中通过 GPU 加速的方式完成渲染计算，并将渲染结果发送至用户端侧进行展示的一类应用程序。在本地桌面场

35

景或浏览器场景等非云渲染场景下，本实施例中的物理机可以是非云服务器，例如，企业机房中的服务器、用户的个人计算机等。在不同的场景中，需要共用物理机上显存的应用程序也是多种多样的，例如，在云渲染场景下，本实施例中的应用程序可以是云桌面程序、云游戏程序、云手机程序或其它云渲染类应用程序；而在非云渲染场景下，本实施例中的应用程序则可以是本地桌面程序或浏览器程序等。本实施例对应用场景不做限定，对物理机在不同应用场景中所承担的工作角色也不做限定，对不同应用场景下支持的应用程序的类型也不做限定。

另外，本实施例中，并不限定物理机内部采用何种方案支持其上运行的应用程序对 GPU 进行调用。例如，可采用虚拟化的方式将物理的 GPU（包括其关联的显存）提供给应用程序使用。又例如，可支持应用程序采用 I/O 的方式直接访问 GPU 等。无论采用何种方案，均不影响本实施例中显存的分配方法的实施。

基于此，参考图 1，本实施例提供的显存的分配方法，可包括：
步骤 100、在拦截到物理机上运行的任一应用程序发起的显存占用请求的情况下，确定应用程序所请求的待占用显存量；

步骤 101、判断待占用显存量和应用程序已占用的显存量之和是否超过应用程序对应的显存占用上限值；

步骤 102、若超过，则拒绝响应显存占用请求。

参考图 2，本实施例中提出在物理机中增设显存过滤组件，并通过显存过滤组件执行本实施例提供的显存的分配方法。

参考图 1，在步骤 100 中，可对物理机上发生的显存占用请求进行拦截。在拦截到物理机上运行的任一应用程序发起的显存占用请求的情况下，可从显存占用请求中解析出该应用程序所请求的待占用显存量。参考图 2，显存过滤组件可监测到物理机上发生的显存占用请求，并对监测到的显存占用请求进行拦截。

图 3 为本公开一示例性实施例提供的一种显存的分配方法的可选实现方式的逻辑示意图。参考图 3，一种可选的拦截实现方案中：显存过滤组件可通过第二预置接口与物理机中的应用渲染组件对接，基于此，显存过滤组件可利用第二预置接口对物理机中的应用渲染组件中的指定接口进行监测，并可在监测到指定接口向物理机中的显卡驱动组件发起显存占用请求的情况下，对监测到的显存占用请求进行拦截。其中，应用渲染组件中的指定接口可以是用于向显卡驱动组件发送显存占用请求的接口。在该拦截实现方案中，为显存过滤组件中设置了第二预置接口，并为该接口配置了接口功能：监测应用渲染组件向显卡驱动组件发送的显存占用请求并拦截监测到的显存占用请求。在传统的显存分配方案中，物理机中的应用渲染组件用于接收物理机中应用程序发起的显存占用请求，并会将显存占用请求转发至物理机中的显卡驱动组件。与传统的渲染过程不同的是，在上述可选的拦截实现方案中，在应用渲染组件和显卡驱动组件之间增设了本实施例提供的显存过滤组件，并且基于前文中提及的为第二预置

接口配置的接口功能，显存过滤组件可利用第二预置接口对应用渲染组件向显卡驱动组件所发起的显存占用请求进行拦截，以避免显存占用请求被直接放行至显卡驱动组件中而被响应。也即是，该拦截实现方式是在应用渲染组件的转发出口处执行拦截操作。

5 值得说明的是，上述的拦截实现方案仅是示例性的，本实施例并不限于此。例如，第二预置接口也可配置为对物理机中应用渲染组件中用于接收应用程序发起的显存占用请求的接口进行监测，并对监测到的显存占用请求进行拦截。也即是，在应用渲染组件的转发入口处执行拦截操作。

其中，本实施例中的拦截是指禁止监测到的显存占用请求被发送至显卡驱动组件。
10 实际应用中，可通过过滤器 Fliter 等技术实现对显存占用请求的拦截，在此不做限定也不做展开详述。

另外，应当理解的是，本实施例中并不限定物理机中所采用的操作系统类型。在不同类型的操作系统中，对应用渲染组件的称谓可能不同，在此不做详尽举例。本实施例中的应用渲染组件可以是指各类操作系统中用于对接应用层并从应用层中接收应用程序发起的渲染任务，以及将渲染任务下发至显卡驱动组件的现有组件。本实施例中
15 中并无需对现有的应用渲染组件进行任何改造，而是利用在物理机中增设的显存过滤组件与现有的应用渲染组件进行交互，以从中拦截物理机中发生的显存占用请求。同样，在不同类型的操作系统中，对显卡驱动组件的称谓也可能不同，在此也不做详尽举例。本实施例中的显卡驱动组件可以是指各类操作系统中用于驱动显卡的现有组件。
20 同样，本实施例中，也并不会对显卡驱动组件进行任何改造，而是利用在物理机中增设的显存过滤组件与现有的显卡驱动组件进行交互，以从中获取执行本实施例提供的显存的分配方法所需的相关信息。参考图 3，本实施例中对物理机中增设的显存过滤组件可设置在应用渲染组件和显卡驱动组件之间，显存过滤组件可分别与应用渲染组件和显卡驱动组件进行交互。

25 与传统的显存分配方案相比，本实施例中，应用程序所发出的显存占用请求不会在被直接传送至显卡驱动组件中，而是会被本实施例中的显卡过滤组件拦截。这样，应用程序将无法再像传统的渲染过程那样，请求多少就能立即占用多少，而是需要等到本实施例中显卡过滤组件的进一步调度。

本实施例中，显存占用请求中可携带有应用程序所请求的待占用显存量。在实际
30 应用中，单个应用程序可发起一次或多次显存占用请求，这些显存占用请求均可在步骤 100 中被拦截到，此后，可按照步骤 101 中的放行条件，对拦截到的显存占用请求无差别的进行条件判断。

继续参考图 1，在步骤 101 中，可为物理机上运行的各个应用程序分别配置显存占用上限值。其中，不同应用程序对应的显存占用上限值可不完全相同。

35 在一种示例性的实现方案中，可由显存过滤组件为各个应用程序动态配置显存占

用上限值。在该示例性的实现方案中：显存过滤组件可响应于达到显存占用上限值配置时机，确定物理机上当前需要共用显存的应用程序的数量和类型；根据物理机上的显存总量以及确定出的数量和类型，为当前需要共用显存的应用程序分别配置显存占用上限值；其中，当前需要共用显存的应用程序分配到的显存占用上限值之和不超过物理机上的显存总量。

在该示例性的实现方案中，显存过滤组件可周期性地执行前述的显存占用上限值配置逻辑。在每一轮的配置过程中，物理机上需要共用显存的应用程序的数量和类型可能会发生变化，相应地，同一应用程序在不同轮的配置过程中所分配到的显存占用上限值就可能是动态变化的。在此不限定上述配置逻辑中具体采用的分配规则，例如，可以按照类型对应用程序进行优先级排序，综合优先级信息和数量信息来分配显存占用上限值，等。在此对分配规则不做限定，也不做更多示例。

应当理解的是，上述提供的显存占用上限值的配置方案仅是示例性的，本实施例并不限于此，本实施例中，还可采用其它方案来实现为应用程序配置显存占用上限值。例如，可根据经验为不同应用程序分配配置显存占用上限值。

基于此，在步骤 101 中，可判断待占用显存量和应用程序已占用的显存量之和是否超过应用程序对应的显存占用上限值。

其中，应用程序已占用的显存量是指在其发起当前的显存占用请求之前已经占用的显存量。而待占用显存量则是其期望通过当前的显存占用请求而请求占用到的显存量。这样，若待占用显存量和应用程序已占用的显存量之和超过应用程序对应的显存占用上限值，则表征响应该显存占用请求后将导致该应用程序占用的显存量超过其对应的显存占用上限值，也即是，会导致该应用程序抢占过量的显存。

在一种可选的实现方式中：可为各个应用程序分别维护显存记录信息，显存记录信息中可包含应用程序的显存占用和/释放记录。这样，可查询步骤 100 中的应用程序所关联的显存记录信息；并基于查询到的显存记录信息，计算应用程序已占用的显存量。

参考图 3，在该可选的实现方式中，显存过滤组件可通过第一预置接口与显卡驱动组件对接，基于此，显存过滤组件可利用第一预置接口，监测物理机中的显卡驱动组件中针对应用程序所进行的显存占用和/或释放操作，以更新为应用程序维护的显存记录信息。值得说明的是，物理机中的显卡驱动组件用于承担显存分配工作。也即是，显卡驱动组件用于响应显存占用请求以为相应的应用程序进行显存占用操作；显卡驱动组件中还可能发生显卡释放操作，以将相应的应用程序所占用的显存进行释放。在该可选的实现方式中，可监测到显卡驱动组件中的显存占用和/或释放操作，从而以应用程序为单位，生成相应的显存占用记录和/或显存释放记录。

其中，显存占用记录可反映应用程序对显存的历史占用总量，而显存释放记录则可反映应用程序对显存的历史释放总量。在此基础上，通过计算应用程序对显存的历

史占用总量与历史释放总量的差值,可获得应用程序已占用的显存量。值得说明的是,实际应用中,应用程序可能并未发生显存释放事件,这种情况下,对应的显存记录信息中将不会有显存释放记录或者显存释放记录为空,同样,若应用程序未成功分配到任何显存,则对应的现有记录信息中也将不会有显存占用记录或者显存占用记录为空。在该可选的实现方式中,按照从显卡驱动组件中实际监测到的显存占用操作及显存释放操作,来为各个应用程序分别维护显存记录信息即可。

可知,在可选的实现方式中,提出了由显存过滤组件为各个应用程序分别维护显存记录信息,这样显存过滤组件可以全面地且及时地掌握各个应用程序的显存占用情况,从而准确确定出应用程序已占用的显存量。这与传统的显存分配方案相比,不再是按照先到先得机制进行显存分配,而是会关注到应用程序的显存占用情况,并限量分配,从而避免先到的应用程序过量占用显存的问题。

继续参考图 1,在步骤 102 中,若确定待占用显存量和应用程序已占用的显存量之和超过应用程序对应的显存占用上限值,则拒绝响应该显存占用请求。

参考图 3,基于显存过滤组件与应用渲染组件之间的对接关系,在步骤 102 中,可向应用渲染组件返回拒绝指令,以禁止将显存占用请求发送至显卡驱动组件;触发应用渲染组件向发起显存占用请求的应用程序返回响应失败通知。示例性地,可采用 STATUS_NO_MEMORY(表示显存容量不足)错误码作为拒绝指令,当然并不限于此。

这样,从应用渲染组件的角度来说,本实施例中显存过滤组件向应用渲染组件返回的拒绝指令一方面可供应用渲染组件获知显存占用请求响应失败,另一方面可触发应用渲染组件向发起该显存占用请求的应用程序返回响应失败通知。从显卡驱动组件的角度来说,其并不会感知到该已被拒绝的显存占用请求的存在。从应用程序的角度来说,则可在接收到响应失败通知后,获知其发起的显存占用请求响应失败。

可选地,本实施例中的拒绝指令可采用与传统显存分配方案中由显卡驱动组件返回给应用渲染组件的拒绝指令一致的指令格式,以使应用渲染组件按照原有通信协议理解步骤 102 中的拒绝指令。这样,从应用渲染组件的角度来说,其并不会感知到显存占用请求被拦截的过程。

在传统的显存分配方案中,应用程序在接收到响应失败通知后,通常会理解为物理机上已经没有剩余显存了,应用程序通常也就不会继续进行显存请求了。与传统的显存分配方案不同的是,本实施例中,支持应用程序针对已经被拒绝的显存占用请求发出重试请求。

为了提高应用程序发出的重试请求的成功率,本实施例中,显卡过滤组件返回给应用渲染组件的拒绝指令可采用指定格式,以区别于传统显存分配方案中由显卡驱动组件返回给应用渲染组件的拒绝指令。这种情况下,特定格式的拒绝指令可用于指示出显存请求过量这一拒绝原因,进一步,显卡过滤组件还可在拒绝指令中携带显存请求量建议,例如,显卡过滤组件可根据该应用程序的显存占用上限值以及已占用的显

存量计算出剩余的可占用显存量，作为显存请求量建议。另外，应用渲染组件在向应用程序反馈的响应失败通知也可采用指定格式，以指示出显存请求过量这一响应失败原因，进一步，也可将前述的显存请求量建议携带在响应失败通知中。而对于应用程序来说，在接收到这类响应失败通知后，可针对被拒绝的显存占用请求发起重试请求。

5 应当理解的是，重试请求的本质依然是显存占用请求。另外，本实施例中，应用程序发起的重试请求中所请求的待占用显存量可以低于其对应的被拒绝的显存占用请求中所请求的待占用显存量。在响应失败通知中携带有显存请求量建议的情况下，应用程序可根据显存请求量建议，配置重试请求中的待占用显存量。例如，配置为低于或等于显存请求量建议中指示的可占用显存量。

10 为此，本实施例中，还可在监测到应用程序在接收到响应失败通知之后发起重试请求的情况下，针对重试请求再次执行确定应用程序所请求的待占用显存量及后续操作，以确定是否允许响应重试请求。也即是，针对应用程序发起的重试请求，也将执行本实施例中前述的步骤 100-102，以确定是否运行响应该重试请求。同样，若该重试请求不满足步骤 101 中的放行条件，则依然会被再次拒绝响应。

15 图 4 为本公开一示例性实施例提供的另一种显存的分配方法的流程示意图。参考图 4，该方法可包括：

步骤 400、在拦截到物理机上运行的任一应用程序发起的显存占用请求的情况下，确定应用程序所请求的待占用显存量；

20 步骤 401、判断待占用显存量和应用程序已占用的显存量之和是否超过应用程序对应的显存占用上限值；若超过，则执行步骤 402；若未超过，则执行步骤 403；

步骤 402、拒绝响应显存占用请求；

步骤 403、将显存占用请求发送至物理机中的显卡驱动组件，以触发显卡驱动组件响应显存占用请求。

其中，步骤 400-步骤 401 可参考前述实施例中的相关描述，在此不做重复赘述。
25 本实施例中基于步骤 403 提供了在待占用显存量和应用程序已占用的显存量之和未超过应用程序对应的显存占用上限值情况下的流程分支，在该流程分支下，显存过滤组件可将从应用渲染组件中拦截的显存占用请求转发至显卡驱动组件；当然，显存过滤组件也可控制应用渲染组件将显存占用请求发送至显卡驱动组件，在此不做限定。

30 可知，本实施例中，若在步骤 101 中确定显存占用请求符合放行条件，则可对显存占用请求进行放行，以正常响应这类显存占用请求。这样，只有不会导致应用程序过滤占用显存的显存占用请求才会被放行，这可保证物理机上运行的每个应用程序所分配到的显存量均不会超过其对应的显存占用上限值，从而可有效避免显存被部分应用程序过量抢占的问题。

35 综上，本实施例中，提出对物理机中应用程序发起的显存占用请求进行拦截，并为物理机中的各个应用程序分别配置显存占用上限值；基于此，在拦截到显存占用请

求的情况下，可判断应用程序本次所请求的待占用显存量和应用程序已占用的显存量之和是否已经超过了应用程序对应的显存占用上限值，若已经超过，则拒绝响应显存占用请求。这样，可基于为不同应用程序配置的显存占用上限值，对物理机中发生的显存占用请求进行过滤，以实现针对不同应用程序所占用的显存进行有效隔离，从而避免显存被部分应用程序过量占用的问题，进而保证共用显存的各个应用程序的正常性能。

图 5 为本公开另一示例性实施例提供的一种显存的请求方法的流程示意图。参考图 5，该方法可适用于物理机中的应用程序，物理机上装配有显存，该方法可包括：

步骤 500、向物理机上的应用渲染组件发起显存占用请求，显存占用请求中携带有待占用显存量；

步骤 501、若接收到应用渲染组件反馈的响应失败通知，则判断响应失败通知是否为指定格式；

步骤 502，若是，则向应用渲染组件发起重试请求，重试请求中携带的待占用显存量低于响应失败的显存占用请求中所携带的待占用显存量；

其中，响应失败通知是由应用渲染组件在接收到物理机上增设的显存过滤组件发送的拒绝指令后反馈至应用程序的，显存过滤组件用于对应用渲染组件中的显存占用请求进行拦截并在确定显存占用请求不符合放行条件的情况下向应用渲染组件发送拒绝指令。另外，这里的放行条件可以指待占用显存量和应用程序已占用的显存量之和是否超过应用程序对应的显存占用上限值。

其中，关于显存过滤组件的相关技术逻辑可参考前述实施例中的相关描述，在此不做赘述。

在一可选的实施例中，一种向应用渲染组件发起重试请求的实现方式，可以是：

若响应失败通知中携带有显存请求量建议，则按照显存请求量建议配置携带在重试请求中的待占用显存量；其中，显存量请求建议为显存过滤组件根据应用程序对应的显存占用上限值和已占用显存量而确定出并提供至应用渲染组件的。

综上，基于本实施例提供的显存的请求方法，物理机中的应用程序可在接收到应用渲染组件反馈的响应失败通知后，调整待占用显存量，并发起重试请求，以请求到显卡过滤组件能够允许的显存量。与传统的显存请求方案相比，应用程序在显存占用请求失败后，不再盲目地放弃请求，而是可基于更加合适的待占用显存量发起重试请求，这样，可使得应用程序能够请求到适量的显存量，进而可实现对不同应用程序进行显存隔离，避免部分应用程序所占用的显存量过多而导致其它应用程序因显存不足而无法正常运行等问题。

图 6 为本公开一示例性实施例提供的一种应用场景的示意图。参考图 6，以物理机上运行的应用程序 A 发起的显存占用请求为例。基于本实施例提供的显存的分配方法和请求方法，大致的处理过程如下：

- 1)应用程序 A 向应用渲染组件发起的显存占用请求,请求的待占用显存量为 2G。
- 2)显存过滤组件从应用渲染组件中拦截到 1)中应用程序 A 所发起的该显存占用请求。
- 3)显存过滤组件还会对显卡驱动组件中的显存占用操作和显存释放操作进行监测,以为应用程序 A 维护了显存记录信息,并通过查询该显存记录信息获知应用程序 A 已占用的显存量为 3G。
- 4)显存过滤组件计算出应用程序 A 的待占用显存量与已占用的显存量之和为 5G,超过了其显存占用上限值 4G。
- 5)显存过滤组件向应用渲染组件返回针对该显存占用请求的
- 10 STATUS_NO_MEMORY 错误码。应用渲染组件向应用程序 A 返回响应失败通知。
- 6)应用程序 A 在过了一段时间后,针对已被拒绝的显存占用请求向应用渲染组件发起了重试请求。重试请求中请求的待占用显存量为 1G。重复步骤 2)。
- 7)显存过滤组件计算出应用程序 A 的待占用显存量与已占用的显存量之和为 4G,未超过了其显存占用上限值 4G。
- 15 8)将重试请求发送至显卡驱动组件,显卡驱动组件为应用程序 A 分配 1G 显存。
- 可知,通过本实施例提供的显存的分配方法和请求方法,可将单个应用程序所占用的显存量限定为不超出其对应的显存占用上限值,从而避免单个应用程序对显存造成过量抢占。而且,基于显存占用上限值,可保证每个应用程序都至少能分配到低于或等于其显存占用上限值的显存量,从而可保证各个应用程序都能正常运行。
- 20 另外,本实施例中,也并不会直接干扰应用程序的启停,而仅是对应用程序发起的显存占用请求进行过滤(例如,应用程序运行在容器中的情况下,本实施例不会直接干扰应用程序所处的容器的暂停或释放),而是将决定权交给应用程序自身,应用程序可针对其被拒绝的显存占用请求决定进行重试还是直接暂停。从而可有效避免因显存的分配不合理而导致应用退出或性能下降等问题。
- 25 需要说明的是,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 101、102 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”
- 30 等描述,是用于区分不同的预置接口等,不代表先后顺序,也不限定“第一”和“第二”是不同的类型。
- 图 7 为本公开又一示例性实施例提供的一种物理机的结构示意图。如图 7 所示,该物理机可包括:存储器 70 和处理器 71。该物理机上还装配有显存。
- 处理器 71,与存储器 70 和显存耦合,用于执行存储器 70 中的计算机程序。
- 35 在一些设计方案中,处理器 71 可用于:

在拦截到所述物理机上运行的任一应用程序发起的显存占用请求的情况下，确定所述应用程序所请求的待占用显存量；

判断所述待占用显存量和所述应用程序已占用的显存量之和是否超过所述应用程序对应的显存占用上限值；

5 若超过，则拒绝响应所述显存占用请求。

在一可选实施例中，处理器 71 还可用于：

查询所述应用程序所关联的显存记录信息，其中，所述显存记录信息中包含所述应用程序的显存占用和/或释放记录；

基于所述显存记录信息，计算所述应用程序已占用的显存量。

10 在一可选实施例中，处理器 71 还可用于：

利用第一预置接口，监测所述物理机中的显卡驱动组件中针对所述应用程序所进行的显存占用和/或释放操作，以更新为所述应用程序维护的显存记录信息。

在一可选实施例中，处理器 71 还可用于：

利用第二预置接口，对所述物理机中的应用渲染组件中的指定接口进行监测；

15 若监测到所述指定接口向所述物理机中的显卡驱动组件发起显存占用请求，则对监测到的显存占用请求进行拦截；

其中，所述应用渲染组件用于接收所述物理机中应用程序发起的显存占用请求。

在一可选实施例中，处理器 71 在拒绝响应所述显存占用请求时，可具体用于：

向所述应用渲染组件返回拒绝指令，以触发所述应用渲染组件向发起所述显存占

20 用请求的应用程序返回响应失败通知。

在一可选实施例中，处理器 71 还可用于：

在监测到所述应用程序在接收到所述响应失败通知之后发起重试请求的情况下，针对所述重试请求再次执行所述确定所述应用程序所请求的待占用显存量及后续操作，以确定是否允许响应所述重试请求。

25 在一可选实施例中，处理器 71 还可用于：

若所述待占用显存量和所述应用程序已占用的显存量之和未超过所述应用程序对应的显存占用上限值，则将所述显存占用请求发送至所述物理机中的显卡驱动组件，以触发所述显卡驱动组件响应所述显存占用请求。

30 在一可选实施例中，所述物理机为云服务器的情况下，所述应用程序包括云渲染类应用程序，所述云渲染类应用程序包括云桌面程序、云游戏程序或云手机程序；所述物理机为非云服务器的情况下，所述应用程序包括本地渲染类应用程序，所述本地渲染类应用程序包括本地桌面程序或者浏览器程序。

在一可选实施例中，处理器 71 还可用于：

35 响应于达到显存占用上限值配置时机，确定所述物理机上当前需要共用显存的应用程序的数量和类型；

根据所述物理机上的显存总量以及确定出的数量和类型，为所述当前需要共用显存的应用程序分别配置显存占用上限值；

其中，所述当前需要共用显存的应用程序分配到的显存占用上限值之和不超过所述物理机上的显存总量。

5 在另一些设计方案中，处理器 71 还可用于：

向所述物理机上的应用渲染组件发起显存占用请求，所述显存占用请求中携带有待占用显存量；

若接收到所述应用渲染组件反馈的响应失败通知，则判断响应失败通知是否为指定格式；

10 若是，则向所述应用渲染组件发起重试请求，所述重试请求中携带的待占用显存量低于响应失败的所述显存占用请求中所携带的待占用显存量；

其中，所述响应失败通知是由所述应用渲染组件在接收到所述物理机上增设的显存过滤组件发送的拒绝指令后反馈至所述应用程序的，所述显存过滤组件用于对所述应用渲染组件中的显存占用请求进行拦截并在确定显存占用请求不符合放行条件的情况

15 下向应用渲染组件发送拒绝指令。

在一可选实施例中，处理器 71 在向所述应用渲染组件发起重试请求时，可具体用于：

若所述响应失败通知中携带有显存请求量建议，则按照所述显存请求量建议配置携带在所述重试请求中的待占用显存量；

20 其中，所述显存量请求建议为所述显存过滤组件根据所述应用程序对应的显存占用上限值和已占用显存量而确定出并提供至所述应用渲染组件的。

进一步，如图 7 所示，该物理机还包括：通信组件 72、电源组件 73 等其它组件。图 7 中仅示意性给出部分组件，并不意味着物理机只包括图 7 所示组件。

25 值得说明的是，上述关于物理机各实施例中的技术细节，可参考前述的方法实施例中的相关描述，为节省篇幅，在此不再赘述，但这不应造成本公开保护范围的损失。

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方法实施例中的各步骤。

30 上述图 7 中的存储器，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计算平台上的操作。这些数据的示例包括用于在计算平台上操作的任何应用程序或方法的指令，联系人数据，电话簿数据，消息，图片，视频等。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（SRAM），电可擦除可编程只读存储器（EEPROM），可擦除可编程只读存储器（EPROM），可编程只读存储器（PROM），只读存储器（ROM），磁存储器，快闪存储器，磁盘或光盘。

35 上述图 7 中的通信组件，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,所述通信组件还包括近场通信(NFC)模块,以促进短程通信。例如,在 NFC 模块可基于射频识别(RFID)技术,红外数据协会(IrDA)技术,超宽带(UWB)技术,蓝牙(BT)技术和其他技术来实现。

上述图 7 中的电源组件,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。

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用存储介质(包括但不限于磁盘存储器、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。

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,并不排除在包括所述要素的过程、方法、商品或者设备中还存在另外的相同要素。

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国家和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。

- 5 以上所述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保护范围之内。

权 利 要 求 书

- 1、一种显存的分配方法，其中，适用于物理机，所述物理机上装配有显存，所述方法包括：
- 5 在拦截到所述物理机上运行的任一应用程序发起的显存占用请求的情况下，确定所述应用程序所请求的待占用显存量；
- 判断所述待占用显存量和所述应用程序已占用的显存量之和是否超过所述应用程序对应的显存占用上限值；
- 若超过，则拒绝响应所述显存占用请求。
- 2、根据权利要求1所述的方法，其中，所述方法还包括：
- 10 查询所述应用程序所关联的显存记录信息，其中，所述显存记录信息中包含所述应用程序的显存占用和/或释放记录；
- 基于所述显存记录信息，计算所述应用程序已占用的显存量。
- 3、根据权利要求2所述的方法，其中，所述方法还包括：
- 利用第一预置接口，监测所述物理机中的显卡驱动组件中针对所述应用程序所进行的显存占用和/或释放操作，以更新为所述应用程序维护的显存记录信息。
- 15 4、根据权利要求1所述的方法，其中，所述方法还包括：
- 利用第二预置接口，对所述物理机中的应用渲染组件中的指定接口进行监测；
- 若监测到所述指定接口向所述物理机中的显卡驱动组件发起显存占用请求，则对监测到的显存占用请求进行拦截；
- 20 其中，所述应用渲染组件用于接收所述物理机中应用程序发起的显存占用请求。
- 5、根据权利要求4所述的方法，其中，拒绝响应所述显存占用请求，包括：
- 向所述应用渲染组件返回拒绝指令，以触发所述应用渲染组件向发起所述显存占用请求的应用程序返回指定格式的响应失败通知；
- 其中，所述拒绝指令中携带有显存请求量建议。
- 25 6、根据权利要求5所述的方法，其中，所述方法还包括：
- 在监测到所述应用程序在接收到所述响应失败通知之后发起重试请求的情况下，针对所述重试请求再次执行所述确定所述应用程序所请求的待占用显存量及后续操作，以确定是否允许响应所述重试请求。
- 7、根据权利要求1所述的方法，其中，所述方法还包括：
- 30 若所述待占用显存量和所述应用程序已占用的显存量之和未超过所述应用程序对应的显存占用上限值，则将所述显存占用请求发送至所述物理机中的显卡驱动组件，以触发所述显卡驱动组件响应所述显存占用请求。
- 8、根据权利要求1所述的方法，其中，所述物理机为云服务器的情况下，所述应用程序包括云渲染类应用程序，所述云渲染类应用程序包括云桌面程序、云游戏程序
- 35 或云手机程序；所述物理机为非云服务器的情况下，所述应用程序包括本地渲染类应

用程序，所述本地渲染类应用程序包括本地桌面程序或者浏览器程序。

9、根据权利要求1所述的方法，其中，所述方法还包括：

响应于达到显存占用上限值配置时机，确定所述物理机上当前需要共用显存的应用程序的数量和类型；

5 根据所述物理机上的显存总量以及确定出的数量和类型，为所述当前需要共用显存的应用程序分别配置显存占用上限值；

其中，所述当前需要共用显存的应用程序分配到的显存占用上限值之和不超过所述物理机上的显存总量。

10 10、一种显存的请求方法，其中，适应于物理机上的应用程序，所述物理机上装有显存，所述方法包括：

向所述物理机上的应用渲染组件发起显存占用请求，所述显存占用请求中携带有待占用显存量；

若接收到所述应用渲染组件反馈的响应失败通知，则判断所述响应失败通知是否为指定格式；

15 若是，则向所述应用渲染组件发起重试请求，所述重试请求中携带的待占用显存量低于响应失败的所述显存占用请求中所携带的待占用显存量；

其中，所述响应失败通知是由所述应用渲染组件在接收到所述物理机上增设的显存过滤组件发送的拒绝指令后反馈至所述应用程序的，所述显存过滤组件用于对所述应用渲染组件中的显存占用请求进行拦截并在确定显存占用请求不符合放行条件的情况下向应用渲染组件发送拒绝指令。
20

11、根据权利要求10所述的方法，其中，向所述应用渲染组件发起重试请求，包括：

若所述响应失败通知中携带有显存请求量建议，则按照所述显存请求量建议配置携带在所述重试请求中的待占用显存量；

25 其中，所述显存量请求建议为所述显存过滤组件根据所述应用程序对应的显存占用上限值和已占用显存量而确定出并提供至所述应用渲染组件的。

12、一种物理机，其中，包括存储器和处理器，所述物理机上装配有显存；

所述存储器用于存储一条或多条计算机指令；

30 所述处理器与所述存储器与所述显存耦合，用于执行所述一条或多条计算机指令，以用于执行权利要求1-9任一项所述的显存的分配方法或者权利要求10-11任一项所述的显存的请求方法。

13、一种存储计算机指令的计算机可读存储介质，其中，当所述计算机指令被一个或多个处理器执行时，致使所述一个或多个处理器执行权利要求1-9任一项所述的显存的分配方法或者权利要求10-11任一项所述的显存的请求方法。

35 14、一种计算机程序，其中，包括计算机指令，所述计算机指令被处理器执行时

实现权利要求 1-9 任一项所述的显存的分配方法或者权利要求 10-11 任一项所述的显存的请求方法。