(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 108984264 B
(45)授权公告日 2022.11.15

(21)申请号 201710407827.8

(22)申请日 2017.06.02

(65)同一申请的已公布的文献号
申请公布号 CN 108984264 A

(43)申请公布日 2018.12.11

(73)专利权人 阿里巴巴集团控股有限公司
地址 英属开曼群岛大开曼资本大厦一座四层847号邮箱

(72)发明人 刘令飞 田双太 龙欣

(74)专利代理机构 北京太合九思知识产权代理有限公司 11610
专利代理师 刘戈

(51)Int.Cl.
G06F 9/455(2006.01)
G06F 9/50(2006.01)

(56)对比文件
CN 102819460 A,2012.12.12
CN 104216783 A,2014.12.17

审查员 杨龙

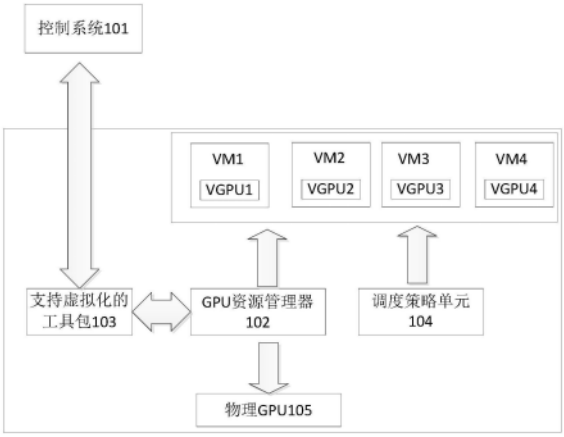
权利要求书3页 说明书15页 附图3页

(54)发明名称

虚拟GPU的实现方法、装置及系统

(57)摘要

本申请公开了一种虚拟GPU的实现方法。所述虚拟GPU的实现方法包括：以在同一个物理GPU上运行的虚拟GPU的资源要求为依据，为各个虚拟GPU分配对应于其资源要求的运行时间片；在所述物理GPU上设置的所有虚拟GPU的运行时间片之和小于等于预设的调度周期上限；以分配给各个虚拟GPU的所述运行时间片为依据，将所述物理GPU的计算资源按照时间片轮询的方式分配给各个虚拟GPU。



CN 108984264 B

1.一种用于实现虚拟GPU的架构系统,其特征在于,包括:控制系统、GPU资源管理器、调度策略单元、支持虚拟化的工具包、虚拟机、虚拟GPU;

所述控制系统,用于查询物理机是否有足够的资源创建预设类型的虚拟GPU;

所述GPU资源管理器,用于管理、分配物理GPU资源、创建虚拟GPU;其中,创建的虚拟GPU具有至少两种确定的规格,并且根据占用物理GPU资源的不同比例作为标准确定虚拟GPU的不同规格;

所述支持虚拟化的工具包,用于作为控制系统与GPU资源管理器交互的通道以及将虚拟GPU加入到虚拟机配置文件中;

所述调度策略单元,用于用动态方式实现对虚拟GPU的调度、增加、删除、升级;

每个虚拟GPU提供给一个所述虚拟机使用;

所述分配物理GPU资源包括:

从调度队列中选取下一个待运行的虚拟GPU;

调用与所述虚拟GPU对应的物理GPU的寄存器接口运行所述虚拟GPU;

当所述虚拟GPU的运行累计时间达到分配给所述虚拟GPU的运行时间片时,通过所述物理GPU寄存器接口,停止所述虚拟GPU运行并保持其状态,然后返回所述从调度队列中选取下一个待运行的虚拟GPU的步骤。

2.一种虚拟GPU的实现方法,其特征在于,包括:

以在同一个物理GPU上运行的虚拟GPU的资源要求为依据,为各个虚拟GPU分配对应于其资源要求的运行时间片;在所述物理GPU上设置的所有虚拟GPU的运行时间片之和小于等于预设的调度周期上限;其中,所述虚拟GPU具有至少两种确定的规格,并且根据占用物理GPU资源的不同比例作为标准确定虚拟GPU的不同规格;

以分配给各个虚拟GPU的所述运行时间片为依据,将所述物理GPU的运算资源按照时间片轮询的方式分配给各个虚拟GPU;

每个虚拟GPU提供给一个虚拟机使用;

所述将所述物理GPU的计算资源按照时间片轮询的方式分配给各个虚拟GPU,包括:

从调度队列中选取下一个待运行的虚拟GPU;

调用与所述虚拟GPU对应的物理GPU的寄存器接口运行所述虚拟GPU;

当所述虚拟GPU的运行累计时间达到分配给所述虚拟GPU的运行时间片时,通过所述物理GPU寄存器接口,停止所述虚拟GPU运行并保持其状态,然后返回所述从调度队列中选取下一个待运行的虚拟GPU的步骤。

3.根据权利要求2所述的虚拟GPU的实现方法,其特征在于,在采用所述虚拟GPU的实现方法的情况下,动态增加物理GPU上运行的虚拟GPU的步骤包括:

判断所述物理GPU是否有足够的剩余时间片及足够的帧缓存空间符合待创建虚拟GPU的运算资源要求;

若是,从未被使用的帧缓存空间链中划分出所述待创建虚拟GPU所需的帧缓存空间;

创建所述待创建虚拟GPU并分配与其对应的虚拟GPU ID;

将所述待创建虚拟GPU加入调度队列。

4.根据权利要求3所述的虚拟GPU的实现方法,其特征在于,所述判断所述物理GPU是否有足够的剩余时间片,包括:

判断应当分配给待创建虚拟GPU的时间片与原有虚拟GPU的时间片之和是否小于或等于所述物理GPU的所述调度周期上限；

若是，判断为有足够的剩余时间片。

5.根据权利要求3所述的虚拟GPU的实现方法，其特征在于，所述判断是否有足够的帧缓存空间的步骤中，所述帧缓存空间是指连续的帧缓存空间。

6.根据权利要求3所述的虚拟GPU的实现方法，其特征在于，所述从未被使用的帧缓存空间链中划分出所述待创建虚拟GPU所需的帧缓存空间，包括：

从未被使用的帧缓存空间链中找到尺寸最接近待创建虚拟GPU的帧缓存空间的尺寸的一段，划分出待创建虚拟GPU所述的帧缓存空间。

7.根据权利要求2所述的虚拟GPU的实现方法，其特征在于，在采用所述虚拟GPU的实现方法的情况下，动态删除物理GPU上运行的虚拟GPU的步骤，包括：

将待删除的虚拟GPU从所述调度队列中移除；

将所述待删除的虚拟GPU的运行时间片按照剩余虚拟GPU对运算资源的要求，按照比例分配给剩余的虚拟GPU。

8.根据权利要求2所述的虚拟GPU的实现方法，其特征在于，在采用所述虚拟GPU的实现方法的情况下，动态升级物理GPU上运行的虚拟GPU的步骤包括：

判断应该分配给升级后的虚拟GPU的运行时间片与其它未升级的虚拟GPU的运行时间片之和是否小于或等于所述调度周期上限；

若是，在所述调度队列中将升级前的虚拟GPU所对应的运行时间片修改为升级后的虚拟GPU所对应的运行时间片。

9.一种虚拟GPU的实现装置，其特征在于，包括：

分配单元，用于以在同一个物理GPU上运行的虚拟GPU的资源要求为依据，为各个虚拟GPU分配对应于其资源要求的运行时间片；在所述物理GPU上设置的所有虚拟GPU的运行时间片之和小于等于预设的调度周期上限；其中，所述虚拟GPU具有至少两种确定的规格，并且根据占用物理GPU资源的不同比例作为标准确定虚拟GPU的不同规格；

调度单元，用于以分配给各个虚拟GPU的所述运行时间片为依据，将所述物理GPU的运算资源按照时间片轮询的方式分配给各个虚拟GPU；

每个虚拟GPU提供给一个虚拟机使用；

所述调度单元，包括：

选取子单元，用于从调度队列中选取下一个待运行的虚拟GPU；

运行子单元，用于调用与所述虚拟GPU对应的物理GPU的寄存器接口运行所述虚拟GPU；

停止子单元，用于当所述虚拟GPU的运行累计时间达到分配给所述虚拟GPU的运行时间片时，通过所述物理GPU寄存器接口，停止所述虚拟GPU运行并保持其状态，然后返回所述从调度队列中选取下一个待运行的虚拟GPU的步骤。

10.根据权利要求9所述的虚拟GPU的实现装置，其特征在于，所述虚拟GPU的实现装置包括增加子单元，用于动态增加物理GPU上运行的虚拟GPU，其包括：

判断时间片及缓存空间子单元，用于判断所述物理GPU是否有足够的剩余时间片及足够的帧缓存空间符合待创建虚拟GPU的运算资源要求；

划分子单元，用于当判断子单元的输出为是时，从未被使用的帧缓存空间链中划分出

所述待创建虚拟GPU所需的帧缓存空间；

创建子单元，用于创建所述待创建虚拟GPU并分配与其对应的虚拟GPU ID；

加入子单元，用于将所述待创建虚拟GPU加入调度队列。

11.根据权利要求10所述的虚拟GPU的实现装置，其特征在于，所述判断时间片及缓存空间子单元，具体用于：

判断应当分配给待创建虚拟GPU的时间片与原有虚拟GPU的时间片之和是否小于或等于所述物理GPU的所述调度周期上限；

若是，判断为有足够的剩余时间片。

12.根据权利要求10所述的虚拟GPU的实现装置，其特征在于，所述足够的帧缓存空间中的帧缓存空间为连续的帧缓存空间。

13.根据权利要求10所述的虚拟GPU的实现装置，其特征在于，所述划分子单元，具体用于：

从未被使用的帧缓存空间链中找到尺寸最接近待创建虚拟GPU的帧缓存空间的尺寸的一段，划分出待创建虚拟GPU所述的帧缓存空间。

14.根据权利要求9所述的虚拟GPU的实现装置，其特征在于，所述虚拟GPU的实现装置包括删除单元，用于动态删除物理GPU上运行的虚拟GPU，包括：

移除子单元，用于将待删除的虚拟GPU从所述调度队列中移除；

分配子单元，用于将所述待删除的虚拟GPU的运行时间片按照剩余虚拟GPU对运算资源的要求，按照比例分配给剩余的虚拟GPU。

15.根据权利要求9所述的虚拟GPU的实现装置，其特征在于，所述虚拟GPU的调度装置包括升级单元，用于动态升级物理GPU上运行的虚拟GPU，包括：

判断子单元，用于判断应该分配给升级后的虚拟GPU的运行时间片与其它未升级的虚拟GPU的运行时间片之和是否小于或等于所述调度周期上限；

修改子单元，用于当判断子单元的输出为是时，在所述调度队列中将升级前的虚拟GPU所对应的运行时间片修改为升级后的虚拟GPU所对应的运行时间片。

16.一种虚拟GPU的实现系统，其特征在于，包括：

处理器；

存储器，用于存储指令，所述指令在被所述处理器读取执行时，执行如权利要求2-8任一项所述的方法。

虚拟GPU的实现方法、装置及系统

技术领域

[0001] 本发明涉及虚拟GPU技术领域，具体涉及一种架构系统、一种虚拟GPU 的实现方法、装置及机器可读介质及系统。

背景技术

[0002] GPU分片虚拟化技术是指将一块物理GPU的资源进行虚拟分片，同时供给一个或多个虚拟机使用的技术，称之为GPU一对多虚拟化技术。每一个GPU 的分片称之为一个虚拟GPU。

[0003] 现有技术下，在将物理GPU进行虚拟化时，一个物理GPU能够分出来的多个虚拟GPU的规格必须是同样的，假设一个物理GPU能支持A,B,C,D四种类型的虚拟GPU。其中A的性能等同于物理GPU的性能，B的性能为物理GPU 性能的1/2，C的性能为物理GPU性能的为1/4，D的性能为物理GPU性能的为1/8，则一个物理GPU只能分成1个类型A的虚拟GPU，或者分成2个B类型的虚拟GPU(每一个占用50％的物理GPU资源)，或者分成4个类型C的虚拟GPU(每一个占用25％的物理GPU资源)，或者分成8个类型D的虚拟GPU。这个划分一旦确定就不能动态更改，除非重新配置物理服务器。也就是说一个物理GPU上只能同时运行一种类型的虚拟GPU，同一台物理服务器上的GPU 实例里用到的虚拟GPU都是同样规格的。

[0004] 现有技术下，控制系统需要事先记录好要每一台物理服务器上的物理GPU 的划分方式和虚拟GPU的数量。按照数据库库存的情况下发创建虚拟机的请求。物理机端，物理GPU到虚拟GPU的划分是在物理机启动的时候完成的，不能在使用的时候，根据需求更改。控制系统端的数据和物理机上的资源划分必须在集群部署之初就静态配置好，并在整个服务过程中保持一致。一个物理GPU 上只能运行同构分片类型的虚拟GPU，不能运行异构分片类型的虚拟GPU，虚拟GPU的性能不能动态调整。

发明内容

[0005] 本申请提供一种架构系统，所述架构系统包括：控制系统、GPU资源管理器、调度策略单元、支持虚拟化的工具包、虚拟机、虚拟GPU；

[0006] 所述控制系统，用于查询物理机是否有足够的资源创建预设类型的虚拟 GPU；

[0007] 所述GPU资源管理器，用于管理、分配物理GPU资源、创建虚拟GPU；

[0008] 所述支持虚拟化的工具包，用于作为控制系统与GPU资源管理器交互的通道以及将虚拟GPU加入到虚拟机配置文件中；

[0009] 所述调度策略单元，用于用动态方式实现对虚拟GPU的调度、增加、删除、升级。

[0010] 本申请另外提供一种虚拟GPU的实现方法，所述方法包括：

[0011] 以在同一个物理GPU上运行的虚拟GPU的资源要求为依据，为各个虚拟 GPU分配对应于其资源要求的运行时间片；在所述物理GPU上设置的所有虚拟 GPU的运行时间片之和小于等于预设的调度周期上限；

[0012] 以分配给各个虚拟GPU的所述运行时间片为依据，将所述物理GPU的运算资源按照

时间片轮询的方式分配给各个虚拟GPU。

[0013] 可选的，在所述物理GPU上设置有多于一个的所述虚拟GPU时，所述虚拟GPU能够设置为具有不同的资源要求的虚拟GPU。

[0014] 可选的，所述将所述物理GPU的计算资源按照时间片轮询的方式分配给各个虚拟GPU，包括：

[0015] 从调度队列中选取下一个待运行的虚拟GPU；

[0016] 调用与所述虚拟GPU对应的物理GPU的寄存器接口运行所述虚拟GPU；

[0017] 当所述虚拟GPU的运行累计时间达到分配给所述虚拟GPU的运行时间片时，通过所述物理GPU寄存器接口，停止所述虚拟GPU运行并保持其状态，然后返回所述从调度队列中选取下一个待运行的虚拟GPU的步骤。

[0018] 可选的，在采用所述虚拟GPU的实现方法的情况下，动态增加物理GPU 上运行的虚拟GPU的步骤包括：

[0019] 判断所述物理GPU是否有足够的剩余时间片及足够的帧缓存空间符合待创建虚拟GPU的运算资源要求；

[0020] 若是，从未被使用的帧缓存空间链中划分出所述待创建虚拟GPU所需的帧缓存空间；

[0021] 创建所述待创建虚拟GPU并分配与其对应的虚拟GPU ID；

[0022] 将所述待创建虚拟GPU加入调度队列。

[0023] 可选的，所述判断所述物理GPU是否有足够的剩余时间片，包括：

[0024] 判断应当分配给待创建虚拟GPU的时间片与原有虚拟GPU的时间片之和是否小于或等于所述物理GPU的所述调度周期上限；

[0025] 若是，判断为有足够的剩余时间片。

[0026] 可选的，所述判断是否有足够的帧缓存空间的步骤中，所述帧缓存空间是指连续的帧缓存空间。

[0027] 可选的，所述从未被使用的帧缓存空间链中划分出所述待创建虚拟GPU所需的帧缓存空间，包括：

[0028] 从未被使用的帧缓存空间链中找到尺寸最接近待创建虚拟GPU的帧缓存空间的尺寸的一段，划分出待创建虚拟GPU所述的帧缓存空间。

[0029] 可选的，在采用所述虚拟GPU的实现方法的情况下，动态删除物理GPU 上运行的虚拟GPU的步骤，包括：

[0030] 将待删除的虚拟GPU从所述调度队列中移除；

[0031] 将所述待删除的虚拟GPU的运行时间片按照剩余虚拟GPU对运算资源的要求，按照比例分配给剩余的虚拟GPU。

[0032] 可选的，在采用所述虚拟GPU的实现方法的情况下，动态升级物理GPU 上运行的虚拟GPU的步骤包括：

[0033] 判断应该分配给升级后的虚拟GPU的运行时间片与其它未升级的虚拟GPU 的运行时间片之和是否小于或等于所述调度周期上限；

[0034] 若是，在所述调度队列中将升级前的虚拟GPU所对应的运行时间片修改为升级后的虚拟GPU所对应的运行时间片。

[0035] 本申请还提供一种虚拟GPU的实现装置,所述装置包括:

[0036] 分配单元,用于以在同一个物理GPU上运行的虚拟GPU的资源要求为依据,为各个虚拟GPU分配对应于其资源要求的运行时间片;在所述物理GPU 上设置的所有虚拟GPU的运行时间片之和小于等于预设的调度周期上限;

[0037] 调度单元,用于以分配给各个虚拟GPU的所述运行时间片为依据,将所述物理GPU的运算资源按照时间片轮询的方式分配给各个虚拟GPU。

[0038] 可选的,所述装置包括,在所述物理GPU上设置有多于一个的所述虚拟 GPU时,所述虚拟GPU能够设置为具有不同的资源要求的虚拟GPU。

[0039] 可选的,所述调度单元,包括:

[0040] 选取子单元,用于从调度队列中选取下一个待运行的虚拟GPU;

[0041] 运行子单元,用于调用与所述虚拟GPU对应的物理GPU的寄存器接口运行所述虚拟GPU;

[0042] 停止子单元,用于当所述虚拟GPU的运行累计时间达到分配给所述虚拟 GPU的运行时间片时,通过所述物理GPU寄存器接口,停止所述虚拟GPU运行并保持其状态,然后返回所述从调度队列中选取下一个待运行的虚拟GPU的步骤。

[0043] 可选的,所述虚拟GPU的实现装置包括增加子单元,用于动态增加物理GPU上运行的虚拟GPU,其包括:

[0044] 判断时间片及缓存空间子单元,用于判断所述物理GPU是否有足够的剩余时间片及足够的帧缓存空间符合待创建虚拟GPU的运算资源要求;

[0045] 划分子单元,用于当判断子单元的输出为是时,从未被使用的帧缓存空间链中划分出所述待创建虚拟GPU所需的帧缓存空间;

[0046] 创建子单元,用于创建所述待创建虚拟GPU并分配与其对应的虚拟GPU ID;

[0047] 加入子单元,用于将所述待创建虚拟GPU加入调度队列。

[0048] 可选的,所述判断时间片子单元,具体用于:

[0049] 判断应当分配给待创建虚拟GPU的时间片与原有虚拟GPU的时间片之和是否小于或等于所述物理GPU的所述调度周期上限;

[0050] 若是,判断为有足够的剩余时间片。

[0051] 可选的,所述足够的帧缓存空间中的帧缓存空间为连续的帧缓存空间。

[0052] 可选的,所述划分子单元,具体用于:

[0053] 从未被使用的帧缓存空间链中找到尺寸最接近待创建虚拟GPU的帧缓存空间的尺寸的一段,划分出待创建虚拟GPU所述的帧缓存空间。

[0054] 可选的,所述虚拟GPU的实现装置包括删除单元,用于动态删除物理GPU 上运行的虚拟GPU,包括:

[0055] 移除子单元,用于将待删除的虚拟GPU从所述调度队列中移除;

[0056] 分配子单元,用于将所述待删除的虚拟GPU的运行时间片按照剩余虚拟 GPU对运算资源的要求,按照比例分配给剩余的虚拟GPU。

[0057] 可选的,所述虚拟GPU的调度装置包括升级单元,用于动态升级物理GPU 上运行的虚拟GPU,包括:

[0058] 判断子单元,用于判断应该分配给升级后的虚拟GPU的运行时间片与其它未升级

的虚拟GPU的运行时间片之和是否小于或等于所述调度周期上限；

[0059] 修改子单元，用于当判断子单元的输出为是时，在所述调度队列中将升级前的虚拟GPU所对应的运行时间片修改为升级后的虚拟GPU所对应的运行时间片。

[0060] 本申请还提供一种系统，所述系统包括：

[0061] 处理器；

[0062] 存储器，用于存储指令，所述指令在被所述处理器读取执行时，执行上述方法。

[0063] 与现有技术相比，本发明具有以下优点：

[0064] 本申请提供了一种虚拟GPU的实现方法，所述方法包括：以在同一个物理 GPU上运行的虚拟GPU的资源要求为依据，为各个虚拟GPU分配对应于其资源要求的运行时间片；在所述物理GPU上设置的所有虚拟GPU的运行时间片之和小于等于预设的调度周期上限；以分配给各个虚拟GPU的所述运行时间片为依据，将所述物理GPU的计算资源按照时间片轮询的方式分配给各个虚拟 GPU。

[0065] 本申请提供的虚拟GPU的实现方法，以在同一个物理GPU上运行的虚拟 GPU的资源要求为依据，为各个虚拟GPU分配对应于其资源要求的运行时间片，按照时间片对虚拟GPU进行调度，本申请提供的方法，不仅支持异构的多个虚拟GPU运行在同一个物理GPU上，而且支持动态创建虚拟GPU，还可以动态删除、升级虚拟GPU，能够动态调整虚拟GPU的性能。

附图说明

[0066] 图1是本申请第一实施例提供的一种架构系统的示意图。

[0067] 图2是本申请第二实施例提供的一种虚拟GPU的实现方法的流程图。

[0068] 图3是本申请第二实施例提供的时间片轮询的调度方式的流程图。

[0069] 图4是本申请第二实施例提供的动态增加物理GPU上运行的虚拟GPU的流程图。

[0070] 图5是本申请第三实施例提供的虚拟GPU的调度装置的示意图。

[0071] 图6是本申请的一种系统的实施例的示意图。

具体实施方式

[0072] 在以下详细说明中，参考构成本文的一部分的附图，其中相同的数字在全文中指相同的部分，并且其中阐述的实施例为本发明的主题可实现的。应当知道可使用其他实施例，并且可实现不脱离本发明范围的结构或逻辑的改变。因此，以下详细说明不是为了限制的意思，并且实施例的范围限定在附带的权利要求和它们的等同替代。

[0073] 各种操作以理解声称的主题最有帮助的方式描述为多个离散的动作或轮流操作。然而，说明书的顺序不应当解释为暗示这些操作为必要的相关顺序。特别地，这些操作不按照呈现的顺序执行。本文描述的操作以描述的实施例不同的顺序来执行。不同的附带操作可被执行和/或在附带的实施例中省略描述的操作。

[0074] 为了本发明的目的，短语“A和/或B”意味着(A)、(B)或(A和B)。为了本发明的目的，短语“A、B和/或C”意味着(A)、(B)、(C)、(A和B)、(A和 C)、(B和C)或(A、B和C)。

[0075] 说明书使用短语“在一个实施例中”或“在实施例中”，其每个都指相同或不同实施例中的一个或多个。而且，术语“包括”、“包含”和“具有”等是同义的，如在本发明实施例中作为参考所使用的那样。

[0076] 如这里所用,术语“电路”、“逻辑”或“模块”指、部分为或包括专用集成电路(ASIC:Application-Specific Integrated Circuit)、电子电路、用于执行一个或多个软件或固件程序的处理器(共用、专用或群)和/或存储器(共用、专用或群)、组合逻辑电路、和/或提供描述的功能的其它合适组件。

[0077] 本申请第一实施例提供了一种架构系统。图1示出了本申请第一实施例提供的一种架构系统的示意图,请参考图1,所述系统包括:控制系统、GPU资源管理器、调度策略单元、支持虚拟化的工具包、虚拟机、虚拟GPU、物理GPU。

[0078] 所述控制系统101,用于查询物理机是否有足够的资源创建预设类型的虚拟 GPU;

[0079] 所述GPU资源管理器102,用于管理、分配物理GPU资源;

[0080] 所述支持虚拟化的工具包103,用于作为控制系统与GPU资源管理器交互的通道以及将虚拟GPU加入到虚拟机配置文件中;

[0081] 所述调度策略单元104,用于用动态方式实现对虚拟GPU的调度、增加、删除、升级。

[0082] 其中,支持虚拟化的工具包可以为Libvert。

[0083] 所述控制系统运行在集群上,所述虚拟机面向使用者提供计算资源,其上运行操作系统和应用程序,每个虚拟GPU即作为某一个虚拟机的一部分向用户提供在虚拟机上实现GPU计算的计算资源。也就是说,所述虚拟GPU在虚拟机层次上是某个虚拟机的一部分;另一方面,从物理实现而言,虚拟GPU是基于物理GPU实现的,一个物理GPU上可以设置多个虚拟GPU。

[0084] 物理GPU实现虚拟GPU的方式,是通过对所述物理GPU的资源在时间、空间等维度的分配使用方式获得的。所述物理GPU的资源包括运算时间片和帧缓存空间;在一个物理GPU上设置某个虚拟GPU的方式,就是将物理GPU的运算时间片分配给该虚拟GPU,并同时对应该虚拟GPU分配相应的帧缓存空间。在同一个物理GPU上可以通过上述方式设置多个虚拟GPU,每个虚拟GPU提供给一个虚拟机使用。

[0085] 本申请提供的架构系统中,所述调度策略单元,就是实现在物理GPU上动态调度、创建、删除、升级虚拟GPU的单元,其具体实现方式可以是一段用于实现上述虚拟GPU相关操作的程序模块,或者是硬件模块。在物理GPU上动态调度所述物理GPU,其实质就是在虚拟GPU已经创建好的情况下,对虚拟 GPU的实现;因此,从狭义上说,可以认为虚拟GPU的调度方法就是对虚拟 GPU的实现。在此基础上,还需要具有创建、删除、升级虚拟GPU的功能,这样才能完全实现虚拟GPU,即一个完整的实现虚拟GPU的方法,应当包括调度、创建、删除以及升级虚拟GPU等各项相关功能。具体的实现过程,在本申请第二实施例中详细说明。

[0086] 上述动态调度、创建、删除、升级虚拟GPU等于虚拟GPU实现相关的功能,均依赖于图1提供的架构系统;上述对虚拟GPU的各项管理功能,即运行在该架构系统上。当然,其中的每一个部分均可以采用不同的具体实现方式。例如,所述支持虚拟化的工具包可以为Libvert,也可以采用其他形式的虚拟化工具包。

[0087] 本申请第二实施例提供了一种虚拟GPU的实现方法。所述实现方法可以运行在图1的架构系统上,也不排除运行在其他的架构系统上。

[0088] 请参考图2,其示出了根据本申请的实施例提供的一种虚拟GPU的实现方法的流程图。以下结合图2进行详细说明。

[0089] 方框201,以在同一个物理GPU上运行的虚拟GPU的资源要求为依据,为各个虚拟

GPU分配对应于其资源要求的运行时间片;在所述物理GPU上设置的所有虚拟GPU的运行时间片之和小于等于预设的调度周期上限。

[0090] 所述GPU,GPU的英文名称为Graphic Processing Unit,GPU中文全称为计算机图形处理器,由1999年NVIDIA公司提出。GPU这一概念也是相对于计算机系统中的CPU而言的,由于人们对图形的需求越来越大,尤其是在家用系统和游戏发烧友,而传统的CPU不能满足现状,因此需要提供一个专门处理图形的核心处理器。GPU作为硬件显卡的“心脏”,地位等同于CPU在计算机系统中的作用。

[0091] 所述物理GPU,指硬件GPU,是显卡中进行图形处理的芯片。

[0092] 所述虚拟GPU,通常指运行在GPU虚拟机上的GPU。虚拟GPU需要占用物理GPU的资源,包括帧缓存空间和运算时间片,用户在使用虚拟GPU时就如同在使用物理GPU,用户感觉不到使用物理GPU与虚拟GPU的差异。

[0093] 所述虚拟GPU的资源要求,指虚拟GPU对物理GPU资源占用的要求,包括时间资源要求和空间资源要求,通常指虚拟GPU对物理GPU运算时间片和帧缓存空间的资源要求。每一种虚拟GPU在建立时会分配固定长度的运算时间片和固定大小的帧缓存空间(Frame Buffer)。

[0094] 如果对虚拟GPU占用的物理GPU资源不做任何限定,允许根据需要随意设置不同资源要求的虚拟GPU,则将可能造成物理GPU的资源最终不能被充分利用。例如,已经设置的虚拟GPU占用了91%的物理GPU资源,剩下的9%的物理GPU资源难以满足任何虚拟机上所要的虚拟GPU的资源要求,则这部分资源就会被浪费。因此,有必要为虚拟GPU设定几个确定的规格,使虚拟GPU 的设置即不像现有技术下只能设置同样规格的虚拟GPU这样没有任何弹性,造成虚拟GPU无法匹配虚拟机的要求;同时,又不会由于过于设置上过于随意,造成物理GPU资源的浪费。

[0095] 具体建立虚拟GPU时,可以根据占用物理GPU资源的不同比例作为标准确定虚拟GPU的不同规格。例如,根据占用物理GPU的比例为100%、50%、 25%、12.5%,设置A、B、C、D四种规格。

[0096] 占用物理GPU资源的比例可以采用所要求的运算时间片占据物理GPU的最长调度周期的比例确定。所谓最长调度周期,就是在该物理GPU上设置的所有虚拟GPU的最长允许轮询时间。如果在一个物理GPU上设置了多个虚拟 GPU,则每个虚拟GPU被轮询一遍所要耗费的时间之和称为总调度周期,该总调度周期不能超过预先设定的最长调度周期。规定最长调度周期,其目的是避免各个虚拟GPU在较长时间内无法获得物理GPU的实际运算资源,影响虚拟 GPU所在虚拟机的用户体验。将该最长调度周期作为物理GPU运算能力的分配基准,就是根据虚拟GPU占据所述最长调度周期的百分比确定不同的虚拟GPU 类型。例如,在前述最长调度周期为40ms的情况下,前面所述的A、B、C、D 四种规格的虚拟GPU,就是分别使用40ms、20ms、10ms、5ms四种运算时间片;在分配时间片的同时,分配相应比例的帧缓存空间,例如,物理GPU具有 4G的帧缓存空间,则A、B、C、D四种规格的虚拟GPU分别分配4G、2G、1G和0.5G。

[0097] 每一个虚拟GPU在创建时确定其规格,在同一个物理GPU上可以设置不同规格虚拟GPU,即实现了异构虚拟GPU的设置。

[0098] 方框202,以分配给各个虚拟GPU的所述运行时间片为依据,将所述物理 GPU的运

算资源按照时间片轮询的方式分配给各个虚拟GPU。

[0099] 所述将所述物理GPU的运算资源按照时间片轮询的方式分配给各个虚拟 GPU,就是将设置在同一个物理GPU上的多个虚拟GPU按照队列调度,轮流使用物理GPU提供的运算资源;以下是一种具体的实现方式,以下结合图3,对该时间片轮询的调度方式进行介绍。所述时间片轮询的调度方式,包括:

[0100] 方框202-1,从调度队列中选取下一个待运行的虚拟GPU。

[0101] 所述虚拟GPU在实际运行中对应于相应的进程,在本具体实现方式下,设置在同一个物理GPU上的各个虚拟GPU的进程在同一个等待队列中等待调度;每个进程包含其对应的虚拟GPU的ID;控制系统通过轮询的办法确定下一个待运行的虚拟GPU,并加载该虚拟GPU的进程,从而实现对待运行的虚拟GPU 的选取。

[0102] 方框202-2,调用与所述虚拟GPU对应的物理GPU的寄存器接口运行所述虚拟GPU。

[0103] 调用某虚拟GPU的进程,就是要将对物理GPU的控制权交给该进程控制。物理GPU的控制权从物理层而言,就是所述虚拟GPU的进程调用其所在物理 GPU的寄存器接口,由当前被调用的该虚拟GPU使用所述寄存器接口,这样,就实质上实现了所述虚拟GPUI对物理GPU的接管和使用,从而实现了运行所述虚拟GPU。所述运行所述虚拟GPU,指物理GPU上的各种资源(存储资源,共享内存,寄存器等)都服务于所述虚拟GPU,使运行在所述虚拟GPU上的程序处于实际被处理的状态中。

[0104] 方框202-3,当所述虚拟GPU的运行累计时间达到分配给所述虚拟GPU的运行时间片时,通过所述物理GPU寄存器接口,停止所述虚拟GPU运行并保持其状态,然后返回所述从调度队列中选取下一个待运行的虚拟GPU的步骤。

[0105] 与方框202-2中所述描述的调用以及运行所述虚拟GPU的过程相反,当所述虚拟GPU的运行累计时间达到分配给所述虚拟GPU的运行时间片后,也就是已经实际占用所述物理GPU的资源到达该虚拟GPU的运算时间片规定的时间后,需要所述虚拟GPU停止对所述物理GPU的实际占用;在退出之前,需要通过寄存器接口停止使用物理GPU,并将刚才运算时间片的计算成果保存到存储器——包括将帧缓存内容保存到分配给该虚拟GPU的帧缓存空间中,还应当包括一些寄存器的内容保存到分配给该虚拟GPU的内存中(通常保存到相应的地址映射寄存器区域),同时,所述虚拟GPU的进程被挂起。上述过程在虚拟机角度看,就是停止所述虚拟GPU的运行并保持所述虚拟GPU的状态,这样,下次调用该虚拟GPU时,就可以直接从该虚拟GPU的上个运算时间片开始运行。

[0106] 当一轮调度周期完成后,进入下一轮调度周期。例如,如果调度队列中有2 个C类型的虚拟GPU,1个D类型的虚拟GPU,C类型的虚拟GPU的时间片为10ms,D类型的虚拟GPU的时间片为5ms,则10+10+5＝25ms为一个调度周期,当本轮25ms调度完成后,每个虚拟GPU都运行了与其对应的时间片相应的时间,进入下一轮调度周期。对于一个物理GPU而言,根据运行在其上的虚拟GPU的数量,其一个调度周期可以小于所述最长调度周期。

[0107] 上述方框201、方框202的说明,仅仅涉及本实施例提供的实现虚拟GPU 的动态调度方法,该调度方法是实现虚拟GPU的基础,该动态调度方法一般在所述调度策略单元104中以软件模块的方式存在,该模块可以称为虚拟GPU调度器。

[0108] 在上述调度方法的基础上,还需要具备动态增加虚拟GPU、动态删除虚拟 GPU以及动态升级虚拟GPU等技术方案配合,才能比较完善的实现虚拟GPU。

[0109] 以下分别介绍动态增加虚拟GPU、动态删除虚拟GPU以及动态升级虚拟 GPU的方法，这些方法均以上述动态调度虚拟GPU为基础，是对本实施例提供的实现虚拟GPU的方案的完善。

[0110] 请参见图4，其为本申请的第一实施例提供的动态增加物理GPU上运行的虚拟GPU的流程图，具体包括框401至框404。

[0111] 判断框401，判断所述物理GPU是否有足够的剩余时间片及足够的帧缓存空间符合待创建虚拟GPU的运算资源要求，若是，执行方框202。

[0112] 当在物理GPU上创建新的虚拟GPU时，首先需要物理GPU上是否有足够的剩余资源来满足待创建虚拟GPU的运算资源要求，即需要考虑判断所述物理 GPU是否有足够的剩余时间片及足够的帧缓存空间。

[0113] 判断所述物理GPU是否有足够的剩余时间片，可以判断应当分配给待创建虚拟GPU的时间片与原有虚拟GPU的时间片之和是否小于或等于所述物理 GPU的所述最大调度周期；若是，判断为有足够的剩余时间片。

[0114] 例如，如果预定的物理GPU的最大调度周期为40ms，原有虚拟GPU的时间片之和30ms，如果待创建虚拟GPU的时间片为10ms，则判断为有足够的剩余时间片，满足创建新的虚拟GPU的时间资源；如果待创建虚拟GPU的时间片为20ms，则待创建虚拟GPU的时间片与原有虚拟GPU的时间片之和为50ms，大于40ms，则判断为没有足够的剩余时间片，不能在物理GPU上创建此类型的待创建虚拟GPU。

[0115] 除了满足有足够的剩余时间片，还需满足有足够的剩余帧缓存空间，只有在两个条件都得到满足时，才能进行待创建虚拟GPU的创建。判断是否有足够的帧缓存空间，只需判断剩余的帧缓存空间的总的容量是否大于或等于待创建虚拟GPU所需的帧缓存空间，足够的帧缓存空间可以是不连续的帧缓存空间，只要各个帧缓存空间的总的容量满足条件即可，优选的，足够的帧缓存空间最好是足够的连续的帧缓存空间，以便为数据的存取带来便利。

[0116] 以使用图1的架构系统动态增加物理GPU上运行的虚拟GPU为例，假设一个物理GPU能支持A,B,C,D四种类型的虚拟GPU。其中A的性能等同于物理GPU的性能，B的性能为物理GPU性能的1/2，C的为1/4，D的为1/8。用户希望创建并增加一个C类型GPU到虚拟机。

[0117] 首先控制系统通过libvirt的接口查询物理GPU是否有足够的运算资源创建 C类型的虚拟GPU；然后libvirt向GPU资源管理器查询是否有足够的运算资源创建C类型的虚拟GPU；GPU资源管理器查询到一台物理GPU，如果该物理GPU的调度周期上限为40ms，在该物理GPU之上建立的原有虚拟GPU的时间片之和为30ms，而C类型虚拟GPU的时间片为10ms，则原有虚拟GPU的时间片之和与C类型虚拟GPU的时间片相加正好为40ms，则判断为有足够的剩余时间片，满足创建新的虚拟GPU的时间资源；在满足时间资源后，GPU资源管理器继续判断此物理GPU是否有足够的剩余帧缓存空间，如果有足够的剩余帧缓存空间，则判断为有足够的剩余帧缓存空间。当然，在标准的实施方式下，会根据虚拟GPU占用的运算时间片相应的分配帧缓存空间，因此，具备足够的剩余时间片也意味着具备足够的帧缓存空间，但是，不排除为虚拟GPU分配资源时，并不绝对的按照为虚拟GPU提供的运算时间片分配相应的帧缓存空间，而是尽量多的先行满足已经建立的虚拟GPU的帧缓存空间要求，这样，就可能出现还有剩余时间片而剩余帧缓存空间不足的情况；另外，也不排除某个物理 GPU的帧缓冲器

出现损坏从而无法为新的虚拟GPU提供足够的帧缓存空间的情况。

[0118] 在判断所述物理GPU有足够的剩余时间片及足够的帧缓存空间符合待创建 C类型虚拟GPU的运算资源要求之后，GPU资源管理器将能够创建的查询结果返回给libvirt，libvirt将查询结果发送给控制系统，控制下发创建虚拟机的请求。

[0119] 需要说明，上述步骤中存在对具体的物理GPU的查询和选择的过程，因为本实施例是假定在云端实施的，有多个可以用于建立虚拟GPU的物理GPU可供使用。

[0120] 方框402，从未被使用的帧缓存空间链中划分出所述待创建虚拟GPU所需的帧缓存空间。

[0121] 当判断框401的输出为是时，从未被使用的帧缓存空间链中划分出所述待创建虚拟GPU所需的帧缓存空间，作为待创建的虚拟GPU的空间资源。

[0122] 优选的，为了充分利用帧缓存空间资源，从未被使用的帧缓存空间链中划分出所述待创建虚拟GPU所需的帧缓存空间，可以从未被使用的帧缓存空间链中找到尺寸最接近待创建虚拟GPU的帧缓存空间的尺寸的一段，划分出待创建虚拟GPU所述的帧缓存空间。

[0123] 在从未被使用的帧缓存空间链中划分出所述待创建虚拟GPU所需的帧缓存空间之前，需要修改实际调度时间(虚拟GPU的时间片之和)，将其修改为原有虚拟GPU的时间片之和与待创建的虚拟GPU的时间片之和。例如，上例中将实际调度时间修改为40ms。

[0124] 仍沿用上述例子，通过本步骤GPU资源管理器从未被使用的帧缓存空间链中划分出所述待创建C类型虚拟GPU所需的帧缓存空间。

[0125] 方框403，创建所述待创建虚拟GPU并为其分配与其对应的虚拟GPU ID。

[0126] 在满足了创建待创建虚拟GPU所需的资源要求并为待创建虚拟GPU划分出了空间资源后，就可以创建新的虚拟GPU，即创建新的虚拟GPU的进程，并分配与其对应的虚拟GPU ID。创建新的虚拟GPU进程时，需要执行一些初期的处理工作，通知相关虚拟机所述虚拟GPU已经准备妥当。

[0127] 方框404，将所述待创建虚拟GPU加入调度队列。

[0128] 在创建完成待创建虚拟GPU后，将虚拟GPU加入调度队列。这样，在根据虚拟GPU调度策略单元执行对虚拟GPU进程的调度的时候，就会在适当的时候通过物理GPU的寄存器接口控制物理GPU的实际使用，从而触发虚拟GPU 切换到该新创建的虚拟GPU。

[0129] 仍沿用上述例子，GPU资源管理器将新创建的C类型虚拟GPU加入调度队列，新创建的C类型虚拟GPU就可以与同一物理机上创建的其他虚拟GPU在调度策略单元提供的调度方式下运行。

[0130] 通过以上介绍，可以看出通过本申请的GPU实现方法，实现了在物理GPU 上动态增加虚拟GPU的功能，而且增加的虚拟GPU可以是异构类型(不同资源要求)的GPU，提高了物理GPU资源的利用率。

[0131] 在采用所述虚拟GPU的调度方法的情况下，动态删除物理GPU上运行的虚拟GPU的步骤，包括：

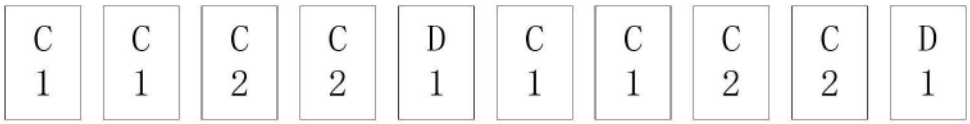
[0132] 将待删除的虚拟GPU从所述调度队列中移除；

[0133] 将所述待删除的虚拟GPU的运行时间片按照剩余虚拟GPU对计算资源的要求，按照比例分配给剩余的虚拟GPU。

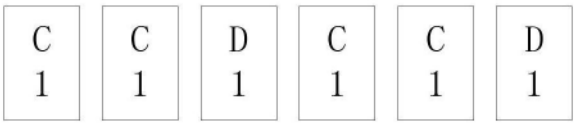
[0134] 在删除虚拟GPU时，只需将待删除的虚拟GPU从所述调度队列中移除，就释放了虚

拟GPU的资源，此时，所述待删除的虚拟GPU的运行时间片实际上相当于按照剩余虚拟GPU对运算资源的要求，按照比例分配给剩余的虚拟 GPU。

[0135] 例如，假设有2个类型C的虚拟GPU和1个类型D的虚拟GPU在同一物理GPU上运行，调度周期上限为40ms，类型C的虚拟GPU的时间片为10ms，类型D的虚拟GPU的时间片为5ms，原有调度队列如下图所示，

[0136] 

[0137] 当动态删除虚拟GPU C2后，调度队列变为下图所示：

[0138] 

……

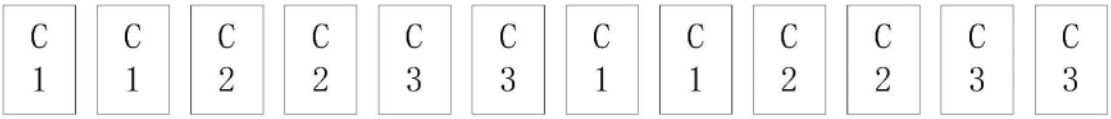
[0139] 其调度周期变为10+5＝15ms，调度周期变小，虚拟GPU C2释放出的10ms 按比例分配到C1和D，每个虚拟GPU被调度的频率增加，运行的总时长变长。

[0140] 在采用所述虚拟GPU的调度方法的情况下，动态升级物理GPU上运行的虚拟GPU的步骤包括：

[0141] 判断应该分配给升级后的虚拟GPU的运行时间片与其它未升级的虚拟GPU 的运行时间片之和是否小于或等于所述调度周期上限；

[0142] 若是，在所述调度队列中将升级前的虚拟GPU所对应的运行时间片修改为升级后的虚拟GPU所对应的运行时间片。

[0143] 仍沿用上面删除前的例子，假设要将D类型的虚拟GPU升级为C类型的虚拟GPU，需要首先判断升级后的虚拟GPU的运行时间片与未升级的虚拟GPU 的运行时间片之和是否小于或等于所述调度周期上限，即：10+10+10＝30ms<40ms，满足升级条件，升级后只需修改虚拟GPU对应的时间片，升级后调度队列为：

[0144] 

[0145] 通过动态升级，充分利用了物理GPU的资源，提高了GPU的运算能力。

[0146] 至此，对本实施例提供的虚拟GPU的调度方法的实施方式进行了描述。通过上述描述可以看出，所述方法支持在同一个物理GPU上实现多个异构虚拟 GPU的运行，还可以动态增加、删除、升级虚拟GPU，只要物理GPU上还有足够的资源，就可以创建一个或多个新的虚拟GPU，在删除某些虚拟GPU后，其所占用的资源可以调整给其他虚拟GPU使用，保证了GPU资源的充分利用，提高了GPU的使用性能。

[0147] 与上述虚拟GPU的实现方法相对应的，本申请还提供了一种虚拟GPU的实现装置。由于所述装置的实施例基本相似于方法的实施例，所以描述得比较简单，相关之处参见方法实施例的部分说明即可。下述描述的装置实施例仅仅是示意性的。所述虚拟GPU的实现装置实施例如下：

[0148] 请参考图5，其示出了根据本申请的第三实施例提供的虚拟GPU的实现装置，所述装置，包括：分配单元501、调度单元502。

[0149] 分配单元501，用于按照在所述物理GPU上设置的虚拟GPU的资源要求，为每个虚拟GPU分配对应于其资源要求的运行时间片；在同一个物理GPU上设置的所有虚拟GPU的运行时间片之和小于等于预设的调度周期上限；

[0150] 调度单元502，用于当所述物理GPU上设置有多于一个的虚拟GPU时，采用时间片轮询的调度方式，将所述物理GPU的运算资源分配给各个虚拟GPU。

[0151] 所述装置包括，在所述物理GPU上设置有多于一个的所述虚拟GPU时，所述虚拟GPU能够设置为具有不同的资源要求的虚拟GPU。

[0152] 可选的，所述调度单元，包括：

[0153] 选取子单元，用于从调度队列中选取下一个待运行的虚拟GPU；

[0154] 运行子单元，用于调用与所述虚拟GPU对应的物理GPU的寄存器接口运行所述虚拟GPU；

[0155] 停止子单元，用于当所述虚拟GPU的运行累计时间达到分配给所述虚拟 GPU的运行时间片时，通过所述物理GPU寄存器接口，停止所述虚拟GPU运行并保持其状态，然后返回所述从调度队列中选取下一个待运行的虚拟GPU的步骤。

[0156] 可选的，所述虚拟GPU的调度装置包括增加子单元，用于动态增加物理 GPU上运行的虚拟GPU，其包括：

[0157] 判断时间片及缓冲区子单元，用于判断所述物理GPU是否有足够的剩余时间片及足够的帧缓存空间符合待创建虚拟GPU的运算资源要求；

[0158] 划分子单元，用于当判断子单元的输出为是时，从未被使用的帧缓存空间链中划分出所述待创建虚拟GPU所需的帧缓存空间；

[0159] 创建子单元，用于创建所述待创建虚拟GPU并获取其对应的虚拟GPU ID；

[0160] 加入子单元，用于将所述待创建虚拟GPU加入调度队列。

[0161] 可选的，所述判断时间片子单元，具体用于：

[0162] 判断应当分配给待创建虚拟GPU的时间片与原有虚拟GPU的时间片之和是否小于或等于所述物理GPU的所述调度周期上限；

[0163] 若是，判断为有足够的剩余时间片。

[0164] 可选的，所述足够的帧缓存空间中的帧缓存空间为连续的帧缓存空间。

[0165] 可选的，所述划分子单元，具体用于：

[0166] 从未被使用的帧缓存空间链中找到尺寸最接近待创建虚拟GPU的帧缓存空间的尺寸的一段，划分出待创建虚拟GPU所述的帧缓存空间。

[0167] 可选的，所述虚拟GPU的调度装置包括删除单元，用于动态删除物理GPU 上运行的虚拟GPU，包括：

[0168] 移除子单元，用于将待删除的虚拟GPU从所述调度队列中移除；

[0169] 分配子单元，用于将所述待删除的虚拟GPU的运行时间片按照剩余虚拟 GPU对运算资源的要求，按照比例分配给剩余的虚拟GPU。

[0170] 可选的，所述虚拟GPU的调度装置包括升级单元，用于动态升级物理GPU 上运行的虚拟GPU，包括：

[0171] 判断子单元，用于判断应该分配给升级后的虚拟GPU的运行时间片与其它未升级的虚拟GPU的运行时间片之和是否小于或等于所述调度周期上限；

[0172] 修改子单元，用于当判断子单元的输出为是时，在所述调度队列中将升级前的虚拟GPU所对应的运行时间片修改为升级后的虚拟GPU所对应的运行时间片。

[0173] 在上述的实施例中，提供了一种架构系统、一种方法以及一种装置，此外，本申请还提供了一种系统，该系统为计算机系统，所述系统实施例如下。

[0174] 请参考图6，其示出了本申请提供的一种系统的实施例的示意图。所述系统 600，可以包括：处理器601，与所述处理器耦合的系统控制单元602，与所述系统控制单元耦合的系统内存(System Memory)603，与所述系统控制单元耦合的非易失存储器(NonVolatile Memory—NVM)或存储设备604，以及与所述系统控制单元耦合的网络接口605。该系统600可以作为一个独立的系统与外部其他单元结合使用，实现对某个或者多个物理GPU上设置虚拟GPU的管理；也可能作为图1所述系统的策略调度单元部分使用。

[0175] 所述处理器601可以包括至少一个处理器，每个处理器可以是单核处理器或者多核处理器。所述处理器601可以包括通用处理器和专用处理器(例如，图形处理器、应用处理器、基带处理器等)的任意组合。在具体实施时，所述处理器601可以被配置为采用不同的实施方式执行图2中所示的方法。

[0176] 所述系统控制单元602可以包括任何相应的接口控制器，为所述处理器601 中的至少一个处理器，和/或，任何与所述系统控制单元602通信的设备或组件，提供接口。

[0177] 所述系统控制单元602可以包括为所述系统内存603提供接口的至少一个内存控制器。所述系统内存603可以用于加载并存储数据和/或指令。所述系统内存603可以包括任何易失性存储器，例如，动态随机存取存储器(dynamic random access memory—DRAM)。

[0178] 所述非易失存储器或存储设备604可以包括至少一个有形的、永久性的计算机可读介质，所述计算机可读介质用于存储数据和/或指令。所述非易失存储器或存储设备604可以包括任何形式的非易失存储器，例如，闪存(flash memory)，和/或，任何非易失存储设备，例如，至少一个硬盘驱动器(hard disk drive—HDD)、至少一个光盘驱动器、和/或、至少一个数字多功能磁盘(digital versatile disk—DVD)驱动器。

[0179] 所述系统内存603和所述非易失存储器或存储设备604可以分别存储有指令程序607的暂时性副本和持久性副本。所述指令程序607存储有执行第二实施例所述方法的指令集合。当所述指令607中的指令被处理器601中的至少一个处理器执行时，使所述系统600执行图2中所示的方法。

[0180] 所述网络接口605可以包括为所述系统600提供无线接口的收发器，所述系统600可以通过无线接口实现跨网络的通信，和/或与其他设备通信。所述网络接口605可以包括任何硬件和/或固件。所述网络接口605可以包括提供多输入、多输出无线接口的多个天线。在具体实施时，所述网络接口605可以为网络适配器、无线网络适配器、电话调制解调器、和/或无线调制解调器。

[0181] 在具体实施时，所述处理器601中的至少一个处理器可以与所述系统控制单元602中的至少一个控制器的控制逻辑封装在一起，形成系统封装(System in Package—SiP)。在具体实施时，所述处理器601中的至少一个处理器可以与所述系统控制单元602中的至少一个控制器的控制逻辑集成在同一个芯片上。在具体实施时，所述处理器601中的至少一个处理器可以与所述系统控制单元602 中的至少一个控制器的控制逻辑集成在同一个芯片上，形成片上系统(System on Chip—SoC)。

[0182] 所述系统600可以进一步包括输入/输出(I/O)设备606。所述输入/输出设备606可以包括供用户与所述系统600进行交互的用户接口，和/或供外围组件和所述系统600交互的外围组件接口。

[0183] 在不同的实施方式中，用户接口可以包括但不局限于：显示器(例如，液晶显示器，触摸屏显示器等)、扬声器、麦克风、至少一个摄像装置(例如相机、和/或摄像机)、闪光灯、以及键盘。

[0184] 在不同的实施方式中，外围组件接口可以包括但不局限于：非易失性存储器端口、音频插孔、以及电源接口。

[0185] 在不同的实施方式中，所述系统600可以部署于个人电脑、移动计算设备等电子设备上，所述移动计算设备可以包括但不局限于：笔记本电脑、平板电脑、移动电话、和/或其他智能设备等。在不同的实施方式中，所述系统600可以包括更多或更少的组件，和/或不同的架构。

[0186] 以下段落描述不同实施例的示例。示例1为一种架构系统，包括：控制系统、GPU资源管理器、调度策略单元、支持虚拟化的工具包、虚拟机、虚拟GPU；所述控制系统，用于查询物理机是否有足够的资源创建预设类型的虚拟GPU；所述GPU资源管理器，用于管理、分配物理GPU资源、创建虚拟GPU；所述支持虚拟化的工具包，用于作为控制系统与GPU资源管理器交互的通道以及将虚拟GPU加入到虚拟机配置文件中；所述调度策略单元，用于用动态方式实现对虚拟GPU的调度、增加、删除、升级。

[0187] 示例2为一种虚拟GPU的实现方法，包括：以在同一个物理GPU上运行的虚拟GPU的资源要求为依据，为各个虚拟GPU分配对应于其资源要求的运行时间片；在所述物理GPU上设置的所有虚拟GPU的运行时间片之和小于等于预设的调度周期上限；以分配给各个虚拟GPU的所述运行时间片为依据，将所述物理GPU的运算资源按照时间片轮询的方式分配给各个虚拟GPU。

[0188] 示例3可以包括如示例2的主题，并且进一步可以指明在所述物理GPU 上设置有多于一个的所述虚拟GPU时，所述虚拟GPU能够设置为具有不同的资源要求的虚拟GPU。

[0189] 示例4可以包括如示例2和3的主题，并且进一步可以指明所述将所述物理GPU的计算资源按照时间片轮询的方式分配给各个虚拟GPU，包括：从调度队列中选取下一个待运行的虚拟GPU；调用与所述虚拟GPU对应的物理GPU 的寄存器接口运行所述虚拟GPU；当所述虚拟GPU的运行累计时间达到分配给所述虚拟GPU的运行时间片时，通过所述物理GPU寄存器接口，停止所述虚拟GPU运行并保持其状态，然后返回所述从调度队列中选取下一个待运行的虚拟GPU的步骤。

[0190] 示例5可以包括如示例2-4的主题，并且进一步可以指明在采用所述虚拟 GPU的实现方法的情况下，动态增加物理GPU上运行的虚拟GPU的步骤包括：判断所述物理GPU是否有足够的剩余时间片及足够的帧缓存空间符合待创建虚拟GPU的运算资源要求；若是，从未被使用的帧缓存空间链中划分出所述待创建虚拟GPU所需的帧缓存空间；创建所述待创建虚拟GPU并分配与其对应的虚拟GPU ID；将所述待创建虚拟GPU加入调度队列。

[0191] 示例6可以包括如示例2-5的主题，并且进一步可以指明所述判断所述物理GPU是否有足够的剩余时间片，包括：判断应当分配给待创建虚拟GPU的时间片与原有虚拟GPU的时间片之和是否小于或等于所述物理GPU的所述调度周期上限；若是，判断为有足够的剩余

时间片。

[0192] 示例7可以包括如示例2-6的主题，并且进一步可以指明所述判断是否有足够的帧缓存空间的步骤中，所述帧缓存空间是指连续的帧缓存空间。

[0193] 示例8可以包括如示例2-7的主题，并且进一步可以指明所述从未被使用的帧缓存空间链中划分出所述待创建虚拟GPU所需的帧缓存空间，包括：

[0194] 从未被使用的帧缓存空间链中找到尺寸最接近待创建虚拟GPU的帧缓存空间的尺寸的一段，划分出待创建虚拟GPU所述的帧缓存空间。

[0195] 示例9可以包括如示例2-8的主题，并且进一步可以指明在采用所述虚拟 GPU的实现方法的情况下，动态删除物理GPU上运行的虚拟GPU的步骤，包括：将待删除的虚拟GPU从所述调度队列中移除；将所述待删除的虚拟GPU 的运行时间片按照剩余虚拟GPU对运算资源的要求，按照比例分配给剩余的虚拟GPU。

[0196] 示例10可以包括如示例2-9的主题，并且进一步可以指明在采用所述虚拟 GPU的实现方法的情况下，动态升级物理GPU上运行的虚拟GPU的步骤包括：判断应该分配给升级后的虚拟GPU的运行时间片与其它未升级的虚拟GPU的运行时间片之和是否小于或等于所述调度周期上限；若是，在所述调度队列中将升级前的虚拟GPU所对应的运行时间片修改为升级后的虚拟GPU所对应的运行时间片。

[0197] 示例11为一种虚拟GPU的实现装置，包括：分配单元，用于以在同一个物理GPU上运行的虚拟GPU的资源要求为依据，为各个虚拟GPU分配对应于其资源要求的运行时间片；在所述物理GPU上设置的所有虚拟GPU的运行时间片之和小于等于预设的调度周期上限；调度单元，用于以分配给各个虚拟GPU 的所述运行时间片为依据，将所述物理GPU的运算资源按照时间片轮询的方式分配给各个虚拟GPU。

[0198] 示例12可以包括如示例11的主题，并且进一步可以指明在所述物理GPU 上设置有多于一个的所述虚拟GPU时，所述虚拟GPU能够设置为具有不同的资源要求的虚拟GPU。

[0199] 示例13可以包括如示例11和12的主题，并且进一步可以指明所述调度单元，包括：选取子单元，用于从调度队列中选取下一个待运行的虚拟GPU；

[0200] 运行子单元，用于调用与所述虚拟GPU对应的物理GPU的寄存器接口运行所述虚拟GPU；停止子单元，用于当所述虚拟GPU的运行累计时间达到分配给所述虚拟GPU的运行时间片时，通过所述物理GPU寄存器接口，停止所述虚拟GPU运行并保持其状态，然后返回所述从调度队列中选取下一个待运行的虚拟GPU的步骤。

[0201] 示例14可以包括如示例11-13的主题，并且进一步可以指明所述虚拟GPU 的实现装置包括增加子单元，用于动态增加物理GPU上运行的虚拟GPU，其包括：判断时间片及缓存空间子单元，用于判断所述物理GPU是否有足够的剩余时间片及足够的帧缓存空间符合待创建虚拟GPU的运算资源要求；划分子单元，用于当判断子单元的输出为是时，从未被使用的帧缓存空间链中划分出所述待创建虚拟GPU所需的帧缓存空间；创建子单元，用于创建所述待创建虚拟GPU 并分配与其对应的虚拟GPU ID；加入子单元，用于将所述待创建虚拟GPU加入调度队列。

[0202] 示例15可以包括如示例11-14的主题，并且进一步可以指明所述判断时间片子单元，具体用于：判断应当分配给待创建虚拟GPU的时间片与原有虚拟 GPU的时间片之和是否小于或等于所述物理GPU的所述调度周期上限；若是，判断为有足够的剩余时间片。

[0203] 示例16可以包括如示例11-15的主题，并且进一步可以指明所述足够的帧缓冲存空间中的帧缓存空间为连续的帧缓存空间。

[0204] 示例17可以包括如示例11-16的主题，并且进一步可以指明所述划分子单元，具体用于：从未被使用的帧缓存空间链中找到尺寸最接近待创建虚拟GPU 的帧缓存空间的尺寸的一段，划分出待创建虚拟GPU所述的帧缓存空间。

[0205] 示例18可以包括如示例11-17的主题，并且进一步可以指明所述虚拟GPU 的实现装置包括删除单元，用于动态删除物理GPU上运行的虚拟GPU，包括：移除子单元，用于将待删除的虚拟GPU从所述调度队列中移除；分配子单元，用于将所述待删除的虚拟GPU的运行时间片按照剩余虚拟GPU对运算资源的要求，按照比例分配给剩余的虚拟GPU。

[0206] 示例19可以包括如示例11-18的主题，并且进一步可以指明所述虚拟GPU 的调度装置包括升级单元，用于动态升级物理GPU上运行的虚拟GPU，包括：判断子单元，用于判断应该分配给升级后的虚拟GPU的运行时间片与其它未升级的虚拟GPU的运行时间片之和是否小于或等于所述调度周期上限；修改子单元，用于当判断子单元的输出为是时，在所述调度队列中将升级前的虚拟GPU 所对应的运行时间片修改为升级后的虚拟GPU所对应的运行时间片。

[0207] 示例20为一种系统，包括：处理器；存储器，用于存储指令，所述指令在被所述处理器读取执行时，执行示例2-10任一项所述的方法。

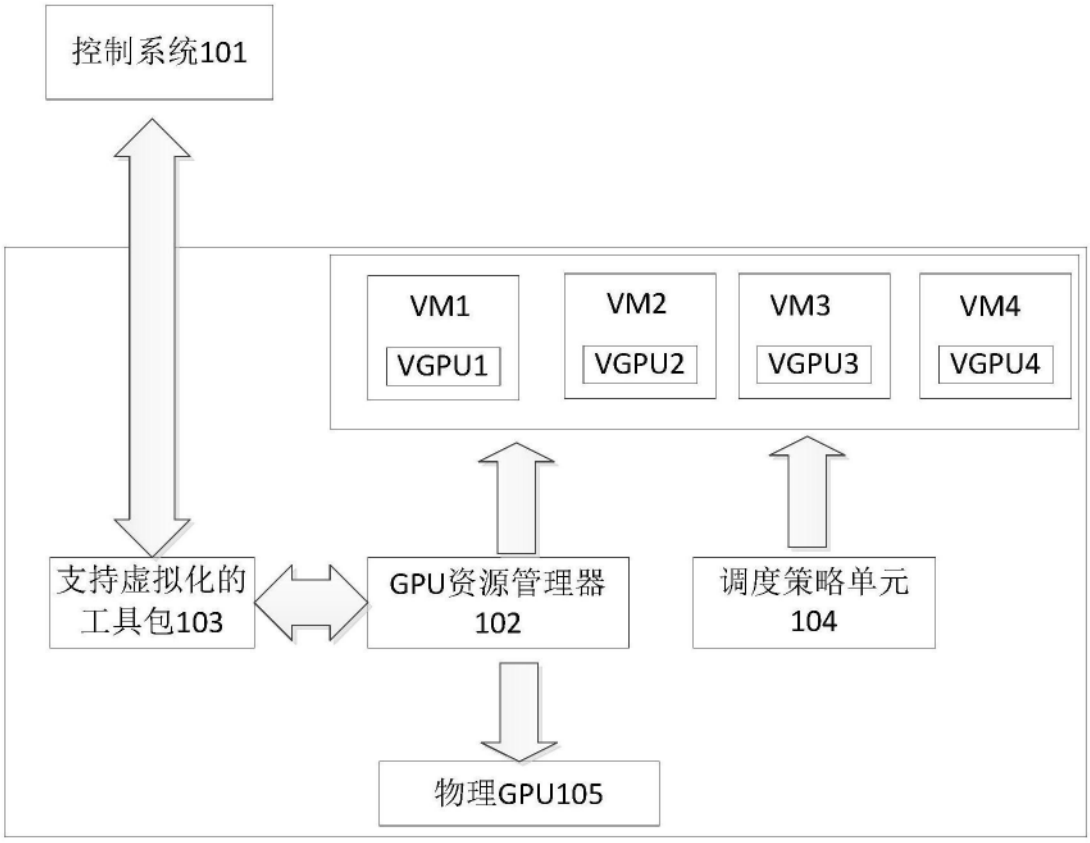

控制系统101
VM1
VGPU1
VM2
VGPU2
VM3
VGPU3
VM4
VGPU4
支持虚拟化的工具包103
GPU资源管理器102
调度策略单元104
物理GPU105

图1

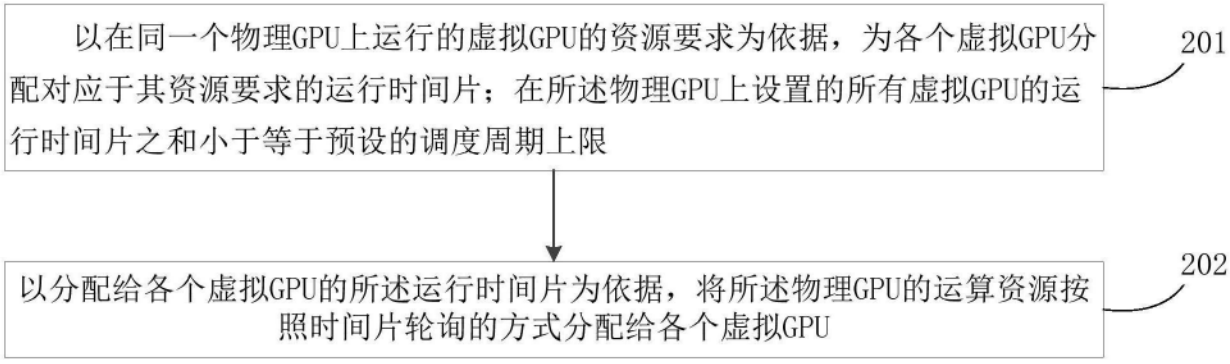

以在同一个物理GPU上运行的虚拟GPU的资源要求为依据，为各个虚拟GPU分配对应于其资源要求的运行时间片；在所述物理GPU上设置的所有虚拟GPU的运行时间片之和小于等于预设的调度周期上限
201
以分配给各个虚拟GPU的所述运行时间片为依据，将所述物理GPU的运算资源按照时间片轮询的方式分配给各个虚拟GPU
202

图2

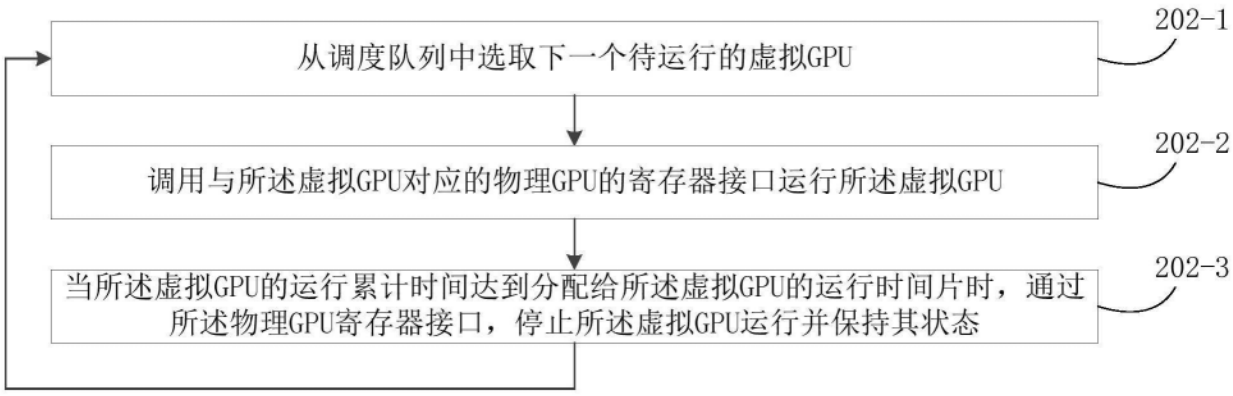

从调度队列中选取下一个待运行的虚拟GPU
202-1
调用与所述虚拟GPU对应的物理GPU的寄存器接口运行所述虚拟GPU
202-2
当所述虚拟GPU的运行累计时间达到分配给所述虚拟GPU的运行时间片时，通过所述物理GPU寄存器接口，停止所述虚拟GPU运行并保持其状态
202-3

图3

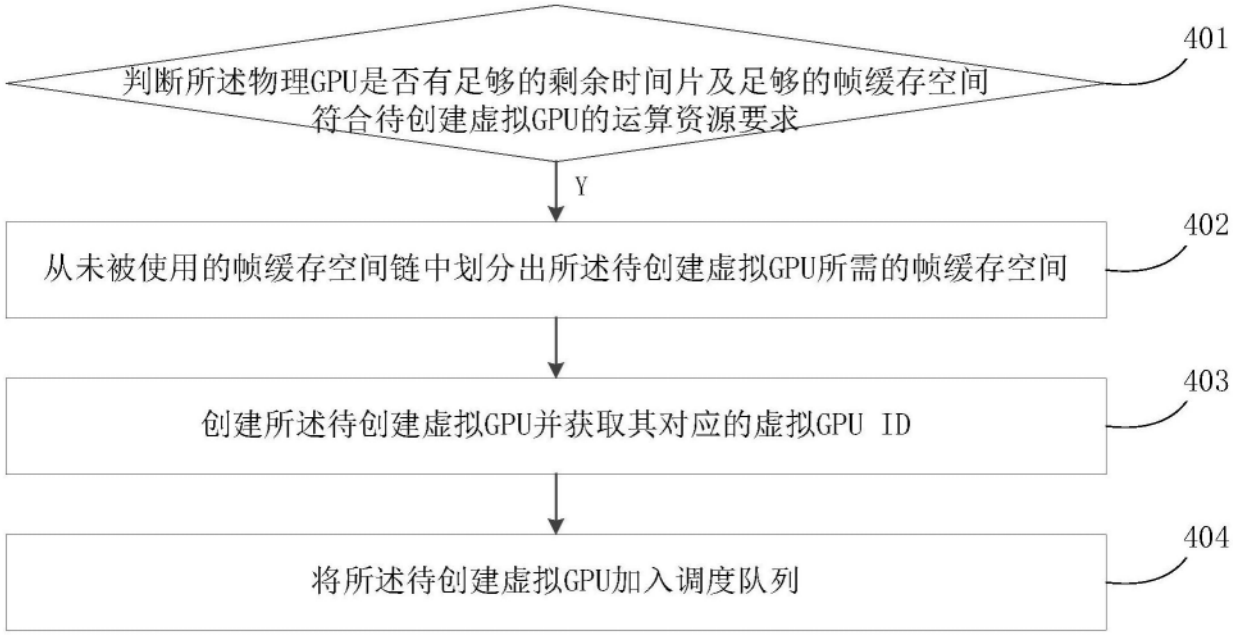

判断所述物理GPU是否有足够的剩余时间片及足够的帧缓存空间符合待创建虚拟GPU的运算资源要求
401
Y
从未被使用的帧缓存空间链中划分出所述待创建虚拟GPU所需的帧缓存空间
402
创建所述待创建虚拟GPU并获取其对应的虚拟GPU ID
403
将所述待创建虚拟GPU加入调度队列
404

图4

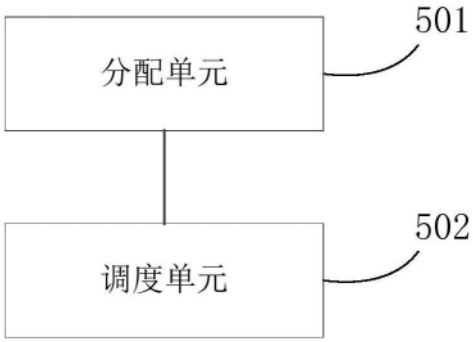

分配单元
501
调度单元
502

图5

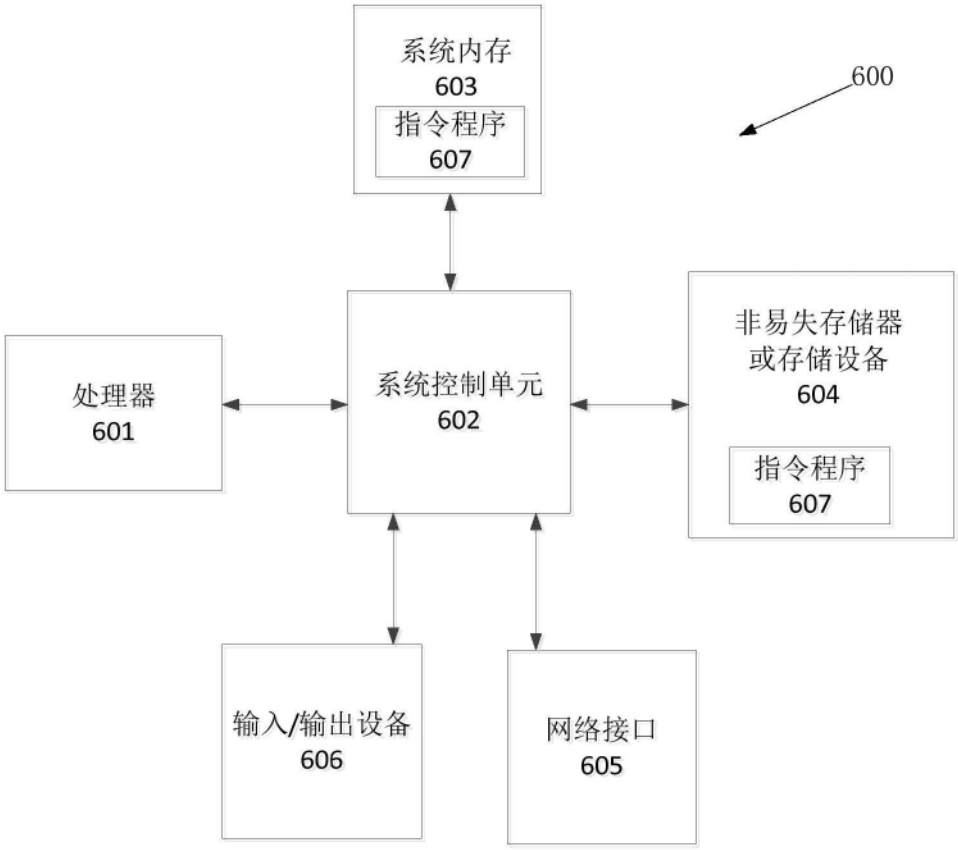

系统内存
603
指令程序
607
600
处理器
601
系统控制单元
602
非易失存储器
或存储设备
604
指令程序
607
输入/输出设备
606
网络接口
605

图6