



(21) 申请号 202210792723.4

(22) 申请日 2022.02.15

(62) 分案原申请数据

202210135055.8 2022.02.15

(71) 申请人 摩尔线程智能科技(北京)有限责任
公司地址 100080 北京市海淀区海淀大街31号2
层209(72) 发明人 李松林 孟平凡 时昊 刘杨
李然月(74) 专利代理机构 北京林达刘知识产权代理事
务所(普通合伙) 11277
专利代理师 刘新宇

(51) Int. Cl.

G06F 12/1009 (2016.01)

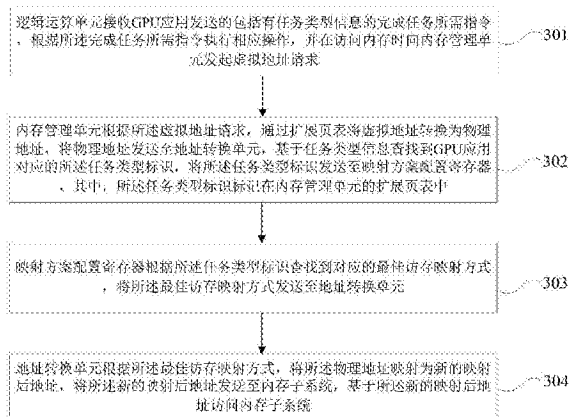
权利要求书4页 说明书11页 附图7页

(54) 发明名称

基于扩展页表的GPU访存自适应优化方法及
装置

(57) 摘要

本发明公开了一种基于扩展页表的GPU访存自适应优化方法及装置,该方法应用于GPU,包括:逻辑运算单元接收GPU应用发送的包括有任务类型信息的完成任务所需指令执行相应操作,并在访问内存时向内存管理单元发起虚拟地址请求;内存管理单元通过扩展页表将虚拟地址转换为物理地址发送至地址转换单元,查找到GPU应用对应的任务类型标识发送至映射方案配置寄存器;映射方案配置寄存器查找到对应的最佳访存映射方式发送至地址转换单元;地址转换单元根据最佳访存映射方式,将物理地址映射为新的地址,将新的地址发送至内存子系统进行访问。本发明在保证访存正确性的同时,实现了在GPU运行时分应用进行针对性的访存优化。



1. 一种GPU访存自适应优化方法,应用于GPU,其特征在于,包括:

逻辑运算单元接收GPU应用发送的包括有任务类型信息的完成任务所需指令,根据所述完成任务所需指令执行相应操作,并在访问内存时向内存管理单元或地址转换单元发起虚拟地址请求;其中,所述虚拟地址请求包括虚拟地址;

映射方案配置寄存器根据所述任务类型信息对应的任务类型标识查找到所述GPU应用对应的最佳访存映射方式,将所述最佳访存映射方式发送至地址转换单元;

地址转换单元根据所述最佳访存映射方式,将所述虚拟地址或所述虚拟地址对应的物理地址映射为映射后地址,所述映射后地址用以访问内存子系统。

2. 如权利要求1所述的GPU访存自适应优化方法,其特征在于,在访问内存时向内存管理单元发起虚拟地址请求的情况下,地址转换单元根据所述最佳访存映射方式,将所述虚拟地址对应的物理地址映射为映射后地址,所述优化方法还包括:

内存管理单元将所述虚拟地址转换为物理地址,将物理地址发送至地址转换单元,并基于任务类型信息查找到GPU应用对应的任务类型标识,将所述任务类型标识发送至映射方案配置寄存器;

地址转换单元将所述映射后地址发送至内存子系统,并基于所述映射后地址访问内存子系统。

3. 如权利要求2所述的GPU访存自适应优化方法,其特征在于,所述内存管理单元将所述虚拟地址转换为物理地址,包括:内存管理单元根据所述虚拟地址请求,通过扩展页表将虚拟地址转换为物理地址,其中,所述任务类型标识标记在内存管理单元的扩展页表中。

4. 如权利要求1所述的GPU访存自适应优化方法,其特征在于,在访问内存时向地址转换单元发起虚拟地址请求的情况下,地址转换单元根据所述最佳访存映射方式,将所述虚拟地址映射为映射后地址,所述优化方法还包括:

地址转换单元将所述映射后地址发送至内存管理单元;其中,所述映射后地址为映射后虚拟地址;

内存管理单元将所述映射后地址转换为物理地址,并基于所述物理地址访问所述内存子系统。

5. 如权利要求1所述的GPU访存自适应优化方法,其特征在于,在访问内存时向地址转换单元发起虚拟地址请求的情况下,地址转换单元根据所述最佳访存映射方式,将所述虚拟地址映射为映射后地址,所述优化方法还包括:

所述地址转换单元根据所述映射后地址,访问所述内存子系统。

6. 如权利要求1至5中任意一项所述的GPU访存自适应优化方法,其特征在于,所述GPU应用对应的最佳访存映射方式按照如下方式确定:

获取GPU应用的GPU访问请求;

利用不同的访存映射方式,将GPU访问请求进行不同的内存地址映射,获得不同映射后的GPU访问请求;

确定不同映射后的GPU访问请求对应的信息熵;

将最大信息熵对应的访存映射方式作为所述GPU应用的最佳访存映射方式。

7. 如权利要求6所述的GPU访存自适应优化方法,其特征在于,确定不同映射后的GPU访问请求对应的信息熵,包括:

根据GPU最大等待队列的长度确定一个时间段；

基于所述时间段确定多个统计时间段；

针对任一映射后的GPU访问请求对应的信息熵，确定每个统计时间段内的所述任一映射后的GPU访问请求在不同访存通道上的分布信息熵；其中，所述不同访存通道与不同的所述内存子系统对应；

根据每个统计时间段内的分布信息熵，确定多个统计时间段内所述任一映射后的GPU访问请求对应的分布信息熵平均值，将所述任一映射后的GPU访问请求对应的分布信息熵平均值，作为所述任一映射后的GPU访问请求对应的信息熵。

8. 如权利要求1至7中任意一项所述的GPU访存自适应优化方法，其特征在于，按照如下方式配置GPU应用对应的最佳访存映射方式：

将所述最佳访存映射方式对应的异或控制向量写入映射方案配置寄存器中，所述异或控制向量的长度与内存地址比特中已被选定进行异或的比特总数相同。

9. 如权利要求8所述的GPU访存自适应优化方法，其特征在于，按照如下方式配置GPU应用对应的最佳访存映射方式：

在将所述最佳访存映射方式对应的异或控制向量写入映射方案配置寄存器中的同时，将GPU应用对应的所述任务类型标识标记在内存管理单元的对应页表中，所述任务类型标识与异或控制向量存在对应关系。

10. 如权利要求1至9中任意一项所述的GPU访存自适应优化方法，其特征在于，还包括：

片上性能寄存器获取GPU应用的内存访问记录，根据所述GPU应用的内存访问记录更新GPU应用的任务类型标识。

11. 一种GPU访存自适应优化装置，应用于GPU，其特征在于，包括：

逻辑运算单元，用于接收GPU应用发送的包括有任务类型信息的完成任务所需指令，根据所述完成任务所需指令执行相应操作，并在访问内存时向内存管理单元或地址转换单元发起虚拟地址请求；其中，所述虚拟地址请求包括虚拟地址；

映射方案配置寄存器，用于根据所述任务类型信息对应的任务类型标识查找到所述GPU应用对应的最佳访存映射方式，将所述最佳访存映射方式发送至地址转换单元；

地址转换单元，用于根据所述最佳访存映射方式，将所述虚拟地址或所述虚拟地址对应的物理地址映射为映射后地址，所述映射后地址用以访问内存子系统。

12. 如权利要求11所述的GPU访存自适应优化装置，其特征在于，在访问内存时向内存管理单元发起虚拟地址请求的情况下，地址转换单元用以根据所述最佳访存映射方式，将所述虚拟地址对应的物理地址映射为映射后地址；

所述内存管理单元，用以将所述虚拟地址转换为物理地址，将物理地址发送至地址转换单元，并基于任务类型信息查找到GPU应用对应的任务类型标识，将所述任务类型标识发送至映射方案配置寄存器；

所述地址转换单元，用以将所述映射后地址发送至内存子系统，并基于所述映射后地址访问内存子系统。

13. 如权利要求12所述的GPU访存自适应优化装置，其特征在于，所述将所述虚拟地址转换为物理地址，包括：根据所述虚拟地址请求，通过扩展页表将虚拟地址转换为物理地址，其中，所述任务类型标识标记在内存管理单元的扩展页表中。

14. 如权利要求11所述的GPU访存自适应优化装置,其特征在于,在访问内存时向地址转换单元发起虚拟地址请求的情况下,地址转换单元用以根据所述最佳访存映射方式,将所述虚拟地址映射为映射后地址;

所述地址转换单元,用以将所述映射后地址发送至内存管理单元;其中,所述映射后地址为映射后虚拟地址;

所述内存管理单元,用以将所述映射后地址转换为物理地址,并基于所述物理地址访问所述内存子系统。

15. 如权利要求11所述的GPU访存自适应优化装置,其特征在于,在访问内存时向地址转换单元发起虚拟地址请求的情况下,地址转换单元用以根据所述最佳访存映射方式,将所述虚拟地址映射为映射后地址;

所述地址转换单元,用以根据所述映射后地址,访问所述内存子系统。

16. 如权利要求11至15中任意一项所述的GPU访存自适应优化装置,其特征在于,所述GPU应用对应的最佳访存映射方式按照如下方式确定:

获取GPU应用的GPU访问请求;

利用不同的访存映射方式,将GPU访问请求进行不同的内存地址映射,获得不同映射后的GPU访问请求;

确定不同映射后的GPU访问请求对应的信息熵;

将最大信息熵对应的访存映射方式作为所述GPU应用的最佳访存映射方式。

17. 如权利要求16所述的GPU访存自适应优化装置,其特征在于,确定不同映射后的GPU访问请求对应的信息熵,包括:

根据GPU最大等待队列的长度确定一个时间段;

基于所述时间段确定多个统计时间段;

针对任一映射后的GPU访问请求对应的信息熵,确定每个统计时间段内的所述任一映射后的GPU访问请求在不同访存通道上的分布信息熵;其中,所述不同访存通道与不同的所述内存子系统对应;

根据每个统计时间段内的分布信息熵,确定多个统计时间段内所述任一映射后的GPU访问请求对应的分布信息熵平均值,将所述任一映射后的GPU访问请求对应的分布信息熵平均值,作为所述任一映射后的GPU访问请求对应的信息熵。

18. 如权利要求11至17中任意一项所述的GPU访存自适应优化装置,其特征在于,按照如下方式配置GPU应用对应的最佳访存映射方式:

将所述最佳访存映射方式对应的异或控制向量写入映射方案配置寄存器中,所述异或控制向量的长度与内存地址比特中已被选定进行异或的比特总数相同。

19. 如权利要求18所述的GPU访存自适应优化装置,其特征在于,按照如下方式配置GPU应用对应的最佳访存映射方式:

在将所述最佳访存映射方式对应的异或控制向量写入映射方案配置寄存器中的同时,将GPU应用对应的所述任务类型标识标记在内存管理单元的对应页表中,所述任务类型标识与异或控制向量存在对应关系。

20. 如权利要求11至19中任意一项所述的GPU访存自适应优化装置,其特征在于,还包括:

片上性能寄存器,用于获取GPU应用的内存访问记录,根据所述GPU应用的内存访问记录更新GPU应用的任务类型标识。

基于扩展页表的GPU访存自适应优化方法及装置

[0001] 本申请是2022年02月15日提交中国专利局、申请号为202210135055.8、发明名称为“基于扩展页表的GPU访存自适应优化方法及装置”的中国专利申请的分案申请。

技术领域

[0002] 本发明涉及访存优化技术领域，尤其涉及基于扩展页表的GPU访存自适应优化方法及装置。

背景技术

[0003] 本部分旨在为权利要求书中陈述的本发明实施例提供背景或上下文。此处的描述不因为包括在本部分中就承认是现有技术。

[0004] 现代计算机内存系统主要有两个优化目标，一是降低内存访问延时，二是提高内存访问带宽。而对GPU内存系统的优化目标则着重于提高访问带宽，而对访问延时有较高容忍度。为此，现代GPU内存系统被划分为可并行访问的多个内存通道，并通过一定的地址-通道映射规则将不同的内存地址映射到不同的内存通道，以提供并行访存能力，提高GPU带宽，如图1所示。通常来说，一系列连续内存访问的内存地址的高位变化较少，而低位变化较为频繁，因此通常选取内存地址中较低的一些位用于选择内存通道。但在实际应用中，由于特定的GPU访存的地址序列，可能导致访问集中于同一个内存通道，进而产生通道访问冲突，如图2所示，当内存为8通道，而访问的地址间隔也为8的整数倍时，所有的请求都会集中在一个通道，而其他7个通道处于空闲状态，导致有效带宽为最大带宽的1/8，这样导致实际的访存带宽大打折扣。

[0005] 为解决地址连续映射到不同通道时导致的如上所述的通道访问冲突问题，在选取若干位(channel bit)用于确定内存通道的同时，用内存地址中的其他bit与channel bit做异或操作之后得到新的地址用于确定内存通道。这一异或操作使得每笔访存请求映射到的内存通道呈现伪随机特性，因而能够在统计意义上达到更好的通道负载均衡。

[0006] 下面详细介绍一下内存地址交织(伪随机交织)方法。

[0007] 表1,表2展示了特定内存排布下的访存冲突情况。表中每一列对应一个内存通道，各列可以并行读写，每对一列中的一个单元进行读写操作需要一个时钟周期。表1为未经过伪随机交织的情况，可见图2所展示的[0,8,16,24,32,40,48]的访问请求均集中在通道1，这样一来需要7个读写周期才能完成访问。而表2为经过伪随机交织，请求被分散到了不同通道，可以并行读写。其中，需要操作最多的为第4列，有三个读写操作(40,48,8)需要完成，故需要三个读写周期完成所有访问。

[0008] 表1为未经过伪随机交织的内存排布情况

[0009]

<u>0</u>	1	2	3	4	5	6	7
<u>8</u>	9	10	11	12	13	14	15
<u>16</u>	17	18	19	20	21	22	23
<u>24</u>	25	26	27	28	29	30	31

<u>32</u>	33	34	35	36	37	38	39
<u>40</u>	41	42	43	44	45	46	47
<u>48</u>	49	50	51	52	53	54	55
56	57	58	59	60	61	62	63

[0010] 表2为经过伪随机交织,请求被分散到了不同通道的内存排布情况

43	37	9	34	44	28	23	30
38	55	53	<u>40</u>	<u>24</u>	21	63	15
1	36	<u>16</u>	49	5	<u>32</u>	60	19
57	59	56	<u>48</u>	3	<u>0</u>	12	13
35	6	27	<u>8</u>	29	25	50	2
51	41	7	46	14	62	47	45
10	33	52	58	39	22	11	31
42	61	17	4	20	18	54	26

[0012] 目前解决内存通道访问冲突的主要办法为上述提到的常用的内存地址交织(伪随机交织)的方式来将GPU原始访存模式映射为在内存通道中分布更加均匀的访问模式,以提高内存通道利用率,进而提高GPU内存系统吞吐率。

[0013] 但随着现代GPU执行的任务类型逐渐增多且逐渐复杂化,固定的内存地址交织方式无法满足所有任务类型的需求。伪随机交织特性使得在大多数情况下,一段时间内的内存访问请求在不同通道之间是相对均衡的,但是由于不同的内存地址交织将内存空间进行了不同的映射,多种内存地址交织方式并存将导致内存地址的重叠,产生访存错误,因而一旦GPU启动,便无法修改内存地址交织方式。

发明内容

[0014] 本发明实施例提供一种基于扩展页表的GPU访存自适应优化方法,用以解决同时存在多种映射,或针对多种应用选取不同映射方式时,内存地址重叠会导致错误,导致现有技术无法扩展到对多种应用自适应选择方案的技术问题,该方法应用于GPU,包括:

[0015] 逻辑运算单元接收GPU应用发送的包括有任务类型信息的完成任务所需指令,根据所述完成任务所需指令执行相应操作,并在访问内存时向内存管理单元发起虚拟地址请求;

[0016] 内存管理单元根据所述虚拟地址请求,通过扩展页表将虚拟地址转换为物理地址,将物理地址发送至地址转换单元,基于任务类型信息查找到GPU应用对应的所述任务类型标识,将所述任务类型标识发送至映射方案配置寄存器,其中,所述任务类型标识标记在内存管理单元的扩展页表中;

[0017] 映射方案配置寄存器根据所述任务类型标识查找到对应的最佳访存映射方式,将所述最佳访存映射方式发送至地址转换单元;

[0018] 地址转换单元根据所述最佳访存映射方式,将所述物理地址映射为新的映射后地址,将所述新的映射后地址发送至内存子系统,基于所述新的映射后地址访问内存子系统。

[0019] 本发明实施例还提供一种基于扩展页表的GPU访存自适应优化装置,用以解决同时存在多种映射,或针对多种应用选取不同映射方式时,内存地址重叠会导致错误,导致现

有技术无法扩展到对多种应用自适应选择方案的技术问题,该装置应用于GPU,包括:

[0020] 逻辑运算单元,用于接收GPU应用发送的包括有任务类型信息的完成任务所需指令,根据所述完成任务所需指令执行相应操作,并在访问内存时向内存管理单元发起虚拟地址请求;

[0021] 内存管理单元,用于根据所述虚拟地址请求,通过扩展页表将虚拟地址转换为物理地址,将物理地址发送至地址转换单元,基于任务类型信息查找到GPU应用对应的所述任务类型标识,将所述任务类型标识发送至映射方案配置寄存器,其中,所述任务类型标识标记在内存管理单元的扩展页表中;

[0022] 映射方案配置寄存器,用于根据所述任务类型标识查找到对应的最佳访存映射方式,将所述最佳访存映射方式发送至地址转换单元;

[0023] 地址转换单元,用于根据所述最佳访存映射方式,将所述物理地址映射为新的映射后地址,将所述新的映射后地址发送至内存子系统,基于所述新的映射后地址访问内存子系统。

[0024] 本发明实施例还提供一种计算机设备,包括存储器、处理器及存储在存储器上并可在处理器上运行的计算机程序,所述处理器执行所述计算机程序时实现上述基于扩展页表的GPU访存自适应优化方法。

[0025] 本发明实施例还提供一种计算机可读存储介质,所述计算机可读存储介质存储有计算机程序,所述计算机程序被处理器执行时实现上述基于扩展页表的GPU访存自适应优化方法。

[0026] 本发明实施例还提供一种计算机程序产品,所述计算机程序产品包括计算机程序,所述计算机程序被处理器执行时实现上述基于扩展页表的GPU访存自适应优化方法。

[0027] 本发明实施例中,与现有技术中同时存在多种映射,或针对多种应用选取不同映射方式时,内存地址重叠会导致错误,导致现有技术无法扩展到对多种应用自适应选择方案的技术方案相比,本发明通过逻辑运算单元接收GPU应用发送的包括有任务类型信息的完成任务所需指令,根据所述完成任务所需指令执行相应操作,并在访问内存时向内存管理单元发起虚拟地址请求;内存管理单元根据所述虚拟地址请求,通过扩展页表将虚拟地址转换为物理地址,将物理地址发送至地址转换单元,基于任务类型信息查找到GPU应用对应的所述任务类型标识,将所述任务类型标识发送至映射方案配置寄存器,其中,所述任务类型标识标记在内存管理单元的扩展页表中;映射方案配置寄存器根据所述任务类型标识查找到对应的最佳访存映射方式,将所述最佳访存映射方式发送至地址转换单元;地址转换单元根据所述最佳访存映射方式,将所述物理地址映射为新的映射后地址,将所述新的映射后地址发送至内存子系统,基于所述新的映射后地址访问内存子系统。本发明通过扩展页表内容,加入任务类型标记,使每个应用可获得定制化的访存映射方式,可以在保证访存正确性的同时,实现在GPU运行时分应用进行针对性的访存优化。

附图说明

[0028] 为了更清楚地说明本发明实施例或现有技术中的技术方案,下面将对实施例或现有技术描述中所需要使用的附图作简单地介绍,显而易见地,下面描述中的附图仅仅是本发明的一些实施例,对于本领域普通技术人员来讲,在不付出创造性劳动的前提下,还可以

根据这些附图获得其他的附图。在附图中：

- [0029] 图1为内存通道示意图；
- [0030] 图2为内存通道出现通道访问冲突的示意图；
- [0031] 图3为本发明实施例中基于扩展页表的GPU访存自适应优化方法示意图一；
- [0032] 图4为本发明实施例中最佳访存映射方式选择操作流程示意图一；
- [0033] 图5为本发明实施例中最佳访存映射方式选择操作流程示意图二；
- [0034] 图6为本发明实施例中不同映射后的GPU访问请求对应的信息熵的操作流程示意图；
- [0035] 图7为本发明实施例中基于扩展页表的GPU访存自适应优化方法示意图二；
- [0036] 图8为本发明实施例中基于扩展页表的GPU访存自适应优化方法示意图三；
- [0037] 图9为本发明实施例中基于扩展页表的GPU访存自适应优化装置结构示意图一；
- [0038] 图10为本发明实施例中基于扩展页表的GPU访存自适应优化装置结构示意图二。

具体实施方式

[0039] 为使本发明实施例的目的、技术方案和优点更加清楚明白，下面结合附图对本发明实施例做进一步详细说明。在此，本发明的示意性实施例及其说明用于解释本发明，但并不作为对本发明的限定。

[0040] 术语解释：

[0041] 内存地址：分为虚拟地址和物理地址两种，一般来说在内存管理单元之前是虚拟地址，内存管理单元之后是物理地址，内存管理单元就负责虚拟地址到物理地址的转换。

[0042] 本申请技术方案中对数据的获取、存储、使用、处理等均符合国家法律法规的相关规定。

[0043] 为解决现有技术中存在的单一地址映射方案的局限性和不同地址映射方案无法共存的问题，本发明提出基于扩展页表的GPU访存自适应优化方法，在保证访存正确性的同时，实现了在GPU运行时分应用进行针对性的访存优化，并实现了软件可配置的映射方式。同时，本方法还扩展了对不同地址映射方案无法共存的支持，提供了应用对访存优化方法的自适应选择。

[0044] 在本发明实施例中，图3为本发明实施例中基于扩展页表的GPU访存自适应优化方法示意图一，如图3所示，该方法应用于GPU，其包括：

[0045] 步骤301：逻辑运算单元接收GPU应用发送的包括有任务类型信息的完成任务所需指令，根据所述完成任务所需指令执行相应操作，并在访问内存时向内存管理单元发起虚拟地址请求；

[0046] 步骤302：内存管理单元根据所述虚拟地址请求，通过扩展页表将虚拟地址转换为物理地址，将物理地址发送至地址转换单元，基于任务类型信息查找到GPU应用对应的所述任务类型标识，将所述任务类型标识发送至映射方案配置寄存器，其中，所述任务类型标识标记在内存管理单元的扩展页表中；

[0047] 步骤303：映射方案配置寄存器根据所述任务类型标识查找到对应的最佳访存映射方式，将所述最佳访存映射方式发送至地址转换单元；

[0048] 步骤304：地址转换单元根据所述最佳访存映射方式，将所述物理地址映射为新的

映射后地址,将所述新的映射后地址发送至内存子系统,基于所述新的映射后地址访问内存子系统。

[0049] 具体的,页表(包含所有的页表项):为兼容在同一个GPU上运行多个程序而不产生内存地址冲突,GPU软件在访问内存时统一使用从0开始的虚拟地址,并通过页表转换成实际的物理地址。具体来说,将物理内存以4KB为单位分成不同的页,通过页表将虚拟地址上的一个4KB页映射到物理地址上的一个4KB地址空间。此外,随着现代计算机内存空间增长,以4KB为单位的页表导致页表项(记录一个页的信息(物理地址,任务类型,以及需要的其他属性)的一条记录)总量过于庞大,进而有了以更大单位划分的页,即为大页。对于大页,同样需要相应的页表,将虚拟地址的大页映射到物理地址的大页。

[0050] 具体的,GPU应用启动时,在向GPU发送完成任务所需指令的同时,还会携带任务类型信息。GPU Core(即逻辑运算单元)根据指令执行操作,并在需要访问内存时,向内存管理单元发起虚拟地址请求。内存管理单元据此将虚拟地址转换为物理地址,并根据任务类型信息查出这笔访存请求对应的任务类型标识(在内存管理单元的对应页表项中存储),传递给映射方案配置寄存器(即GPU的片上寄存器),查找出对应的地址映射方案,传递给地址转换模块。地址转换模块接收到物理地址和映射方案后,按照规则将其映射为新的映射后地址,并发送给内存子系统,然后基于新的映射后地址进行内存子系统的访问。

[0051] 在GPU程序启动时,将根据该程序的应用类型为其分配一个任务类型标识(例如为科学计算型应用分配类型值0,为图形渲染应用分配类型值1等,亦可为另一种与前者访存模式差异较大的图形渲染应用分配类型值2),并记录在分配给该程序的页表项中。当程序发起访存请求时,就会在将虚拟地址通过页表项翻译为物理地址的同时,取出任务类型值传入地址伪随机交织模块,将应用原始访存模式映射为优化后该应用的最佳访存模式。

[0052] 具体的,访存映射方式(方法)对应于一种内存地址映射规则,例如缓存内部的访问是使用组、路的寻址方式,当然组和路也是通过实际内存地址计算得来的。

[0053] 在本发明实施例中,如图4所示,所述GPU应用对应的最佳访存映射方式按照如下方式确定:

[0054] 步骤401:获取GPU应用的GPU访问请求;

[0055] 步骤402:利用不同的访存映射方式,将GPU访问请求进行不同的内存地址映射,获得不同映射后的GPU访问请求;

[0056] 步骤403:确定不同映射后的GPU访问请求对应的信息熵;

[0057] 步骤404:将最大信息熵对应的访存映射方式作为所述GPU应用的最佳访存映射方式。

[0058] 具体的,访存映射方式包括但不限于以异或控制向量为基础的映射方式,还可以是将地址乘上一个质数再按照通道数取模的方法,则对应于不同的质数类型。

[0059] 例如,对于一个原始访问集中在通道1,2上的应用,可以通过对选择通道的bit的高位进行扰动,使一部分请求分布到3,4通道。相应的,如果集中在1,3通道,则可扰动低bit使一部分请求分不到2,4通道。

[0060] 具体的,如图5所示,步骤403确定不同映射后的GPU访问请求对应的信息熵,包括:

[0061] 步骤501:根据GPU最大等待队列的长度确定一个时间段;

[0062] 步骤502:基于所述时间段确定多个统计时间段;

[0063] 步骤503:确定每个统计时间段内的GPU访问请求在不同访存通道上的分布信息熵;

[0064] 步骤504:根据每个统计时间段内的分布信息熵确定多个统计时间段的分布信息熵平均值,将所述分布信息熵平均值作为映射后的GPU访问请求对应的信息熵。

[0065] 具体的,如图6所示,针对不同GPU应用,需要找到相应的优化方法。对于实际应用A(或一组特定类型的应用),将应用运行在GPU仿真器(研发阶段)或GPU芯片(后续优化)上,并抓取GPU的内存访问序列(访问请求记录)。此后,对所有可能的访存映射方式(体现为不同的异或控制向量),基于信息熵评估每种映射方式映射后的内存访问请求在通道上的分布均匀性,并从中选取分布最均匀(也即信息熵最高)的一种映射,得到针对应用A(或一组特定类型应用)的最佳访存映射方式。

[0066] 下面介绍一下信息熵。

[0067] 信息熵E是衡量一组消息携带信息的量,其公式为:

[0068] $E = \sum (-p_i \times \log_2(p_i))$ 。

[0069] 即消息传输中每种可能的消息i的概率 p_i 的负对数之和。

[0070] 例如1bit传输,假如传输数据为0和为1的概率各位1/2,则 $E = 1/2 + 1/2 = 1$,也即每次传输带来了1bit的信息。相反,假如传输数据一直为0或一直为1,则 $E = 0$,也即该数据传输并不携带信息量。一般而言,当传输的数据不确定性越强时,一组接收到的确定性的结果能够携带的信息量越多。

[0071] 下面描述一下基于信息熵的访存分布均匀性评估。

[0072] 记录的每一条访存请求为(t,a),其中t为访存请求发起的时间,a为访存请求对应的地址(此处仅选取用于选择通道的部分)。将根据GPU最大等待队列(一项GPU内部配置。GPU允许在内存请求得到回复前继续下发多个内存请求,但这一数量是有限制的)的长度决定一个时间段T,按照 $0 \sim 1T, 1T \sim 2T \dots (N-1)T \sim NT$ 分段统计每个时段内的GPU访存请求在不同通道上的分布,并计算这一分布的熵,最后再对作为这一访存序列中的所有时段的信息熵取平均作为该访存序列均匀性的评估。其中,NT=纳入统计的总时间长度,与实际应用运行时间并无直接联系,但不可能超过实际应用运行时间。

[0073] 对于每一种映射方式f,均将原始请求(t,a)的序列映射为由(t,a_f)组成的新序列。将通过计算每一种映射方式带来的新序列的信息熵来评估一个访存映射方式的优劣(映射后序列信息熵越高则映射方式越好)。

[0074] 对于本发明提到的内存地址序列的信息熵,统计一个固定时间窗口内,每个内存地址比特(如bit0,一个内存地址通常包含多个比特)组成的序列(如bit0_t0,bit0_t1,bit0_t2...),进而得到每个比特在这个时间窗内为0和为1的概率,并应用上述公式计算比特序列的信息熵。

[0075] 例如,一个时间窗口内记录了4笔内存地址的操作(0000,0001,0011,0111),则bit0构成的序列为(0,1,1,1),bit1构成的序列为(0,0,1,1),bit2构成的序列为(0,0,0,1),bit3构成的序列为(0,0,0,0)。由此计算出来bit0的信息熵 $E_0 = -p_0(\log_2(p_0) - p_1(\log_2(p_1) = -0.25 \times \log_2(0.25) + 0.75 \times \log_2(0.75) = 0.811$,同理可得 $E_1 = 1, E_2 = 0.811, E_3 = 0$ 。

[0076] 在本发明实施例中,如图7所示,该方法还包括:

[0077] 步骤701:映射方案配置寄存器中配置有GPU应用对应的最佳访存映射方式。

[0078] 具体的,当映射方案配置有所更新,或特定用户需要进行定制化的优化时,在用户使用GPU时,可由GPU优化软件(软件方式)将得到的最佳映射方案写入到片上寄存器中,实现GPU映射方案的动态更新与定制。即修改映射方案配置寄存器的值(即异或控制向量),实现GPU映射方案的动态更新与定制。

[0079] 本发明通过软件可访问的寄存器配置通道,实现产品发布后通过软件进行内存映射方案的更新。

[0080] 具体的,当映射方案配置有所更新,或特定用户需要进行定制化的优化时,还可以通过GPU驱动或固件(硬件方式)(GPU驱动或固件的具体包括哪些取决于GPU的具体实现方式)将GPU应用对应的最佳访存映射方式动态配置到映射方案配置寄存器,实现GPU映射方案的动态更新与定制。

[0081] 实施例1单个伪随机映射方案的配置方法。

[0082] 当配置一个特定的伪随机映射方案时,可通过固件或驱动将对应的异或控制向量(根据最佳访存映射方式来确定)a写入片上映射方案配置寄存器中,地址转换模块即可根据片上映射方案配置寄存器写入的异或控制向量将原始物理地址转换为映射后实际用于访问内存子系统的地址。地址转换模式应用内存控制向量进行地址转换的具体方法如下:

[0083] 定义对于一个内存地址B,其由多个比特组成,定义构成这一地址的多个比特从低到高分别为 b_1, b_2, \dots, b_n 。

[0084] 伪随机映射通过选取内存地址中的若干比特 b_i 与将要用于选取内存通道(即内存地址)的比特 b_c (channel bit)做异或操作,得到新的用于选取内存通道的比特(channel bit) b_c' :

[0085] $b_c' = b_c \oplus b_1 \oplus b_2 \oplus b_3 \dots$;

[0086] 但对于不同的应用, b_i 在内存地址中的位置也不尽相同。例如,对于一种映射方案,可以有

[0087] $b_c' = b_c \oplus b_1 \oplus b_4 \oplus b_6 \dots$;

[0088] 而对于另一种方案,则可能是

[0089] $b_c' = b_c \oplus b_2 \oplus b_3 \oplus b_7 \dots$;

[0090] 为实现动态配置,本申请通过异或控制向量a来控制要参与异或的比特,异或控制向量a的长度与所有可能参与异或的bit总数相同。在进行伪随机映射时,首先将异或向量a与所有可能参与异或的bit做与操作,再将结果与channel bit(b_c)做异或操作,由此实现了当a中某一个元素 $a_i = 0$ 时,对应的 $a_i \& b_i$ 也为0,不会改变异或结果,也就不会影响伪随机交织结果。其中,所有可能参与异或的bit的选择方式是任意的,如随机指定,或尝试所有可能的bit组合选择最优,或根据实际访存模式选择有变化的bit等(例如可能在有的应用的访存模式下,某些bit完全没有变化,则他们参与异或时毫无作用)。

[0091] $b_c' = b_c \oplus (b_1 \& a_1) \oplus (b_2 \& a_2) \oplus \dots$ 。

[0092] 实施例2多个伪随机映射方案的配置方法。

[0093] 对于多个伪随机映射方案,在GPU中用于存储异或控制向量的寄存器扩充至可容纳N组异或控制向量(与最佳访存映射方式对应),同时对每个大页对应的页表项增加一个任务类型标识,每个页表项根据其在页表中记录的任务类型标识在寄存器中查找到对应的

异或控制向量(寄存器中存储有任务类型标识和异或控制向量的对应关系),并进行地址伪随机交织。由于页的划分已经提供了不同页之间的内存空间隔离,每个页可以在自己对应的内存空间自由应用与其他页不同的内存地址映射方案,而不会互相影响。如表3所示,三个字体对应三个不同的映射方案,粗体为无映射,标准字体为纯随机映射,斜体为列交换映射。页1执行任务A选择映射方式1,页2和页4执行任务B(或执行同类任务B,D)选择了映射方式2,页3执行任务C,选择了映射方式3。他们都能够得到针对自己执行任务类型而进行的专有优化。同时,随机交织只会在每个大页对应的基本块内进行,确保了不同的映射方式不会产生跨内存块的存储空间重叠,以确保数据安全。

[0094] 表3不同的内存地址映射方案

[0095]

0	1	27	3	<i>5</i>	<i>4</i>	6	7
8	9	10	59	<i>13</i>	<i>12</i>	23	15
16	17	34	19	<i>21</i>	<i>20</i>	22	30
24	25	26	2	<i>29</i>	<i>28</i>	14	38
32	33	18	35	<i>37</i>	<i>36</i>	47	46
40	41	42	50	<i>45</i>	<i>44</i>	31	39
48	49	58	51	<i>53</i>	<i>52</i>	63	55
56	57	43	11	<i>61</i>	<i>60</i>	62	54

[0096] 在本发明实施例中,如图8所示,该方法还包括:

[0097] 步骤801:片上性能寄存器获取GPU应用的内存访问记录,根据所述GPU应用的内存访问记录更新GPU应用的任务类型标识。

[0098] 具体的,应用通过页表中分配的任务类型标识选择对应的异或控制向量。对于常用任务,会对其进行针对性优化,从而可通过驱动程序为其分配预先优化选择好的任务类型标识,进而选择对应的异或控制向量,达到性能最优化。但对于非常用任务类型,或尚未进行针对性优化的任务,本发明还可以根据片上性能计数器收集的反馈结果,自适应地根据应用需求与性能表现,在下一次启动时选择更加合理的内存映射方案(体现为选择另一种更加合适的任务类型标识),即本发明还支持通过片上性能计数器获取的芯片实时工作负载情况而动态学习调整内存映射方式。具体而言,此功能利用GPU芯片中存在的性能计数器(如带宽、延时、功耗等统计信息),动态分析应用实际带宽需求,并记录在片上非易失存储单元,或者记录在计算机硬盘存储空间中,从而在下一次访问时,由驱动读出相应信息,并按记录为应用选择合适的映射方案(体现为分配恰当的任务类型标识)。

[0099] 此方法区别于多个伪随机映射方案的配置方法的是,多个伪随机映射方案的配置方法描述的为人工手动为每个应用分配类型标识,但是GPU应用众多,不可能一一手动优

化,因此通过此方法可以自动调整应用适用的场景。一个例子是,对于计算密集型应用,片上性能计数器检测到其内存访问请求并不密集,此时可以选择不进行映射,或者映射到更少的通道,以减少活动通道数量,降低整体功耗。另一个例子是,有些特定应用已经预先做好了通道访问的均匀化,此时对其再进行伪随机映射,反而会破坏原本精心优化的访存模式,进而导致性能劣化。此时即可通过统计GPU原始访存请求平衡性,选择不进行任何映射的方案,从而保持原始应用优化的收益。以上两个例子仅为特定的具体应用,实际用途包括但不限于上述两种情况。

[0100] 此外,还可以通过配置寄存器来实现伪随机交织、映射方案动态选择的开关,从而实现更加灵活的GPU内存-通道映射。

[0101] 本发明实施例中还提供了一种基于扩展页表的GPU访存自适应优化装置,如下面的实施例所述。由于该装置解决问题的原理与基于扩展页表的GPU访存自适应优化方法相似,因此该装置的实施可以参见基于扩展页表的GPU访存自适应优化方法的实施,重复之处不再赘述。

[0102] 图9为本发明实施例中基于扩展页表的GPU访存自适应优化装置结构示意图,如图9所示,该装置应用于GPU,包括:

[0103] 逻辑运算单元(即GPU Core),用于接收GPU应用发送的包括有任务类型信息的完成任务所需指令,根据所述完成任务所需指令执行相应操作,并在访问内存时向内存管理单元发起虚拟地址请求;

[0104] 内存管理单元,用于根据所述虚拟地址请求,通过扩展页表将虚拟地址转换为物理地址,将物理地址发送至地址转换单元,基于任务类型信息查找到GPU应用对应的所述任务类型标识,将所述任务类型标识发送至映射方案配置寄存器,其中,所述任务类型标识标记在内存管理单元的扩展页表中;

[0105] 映射方案配置寄存器,用于根据所述任务类型标识查找到对应的最佳访存映射方式,将所述最佳访存映射方式发送至地址转换单元;

[0106] 地址转换单元,用于根据所述最佳访存映射方式,将所述物理地址映射为新的映射后地址,将所述新的映射后地址发送至内存子系统,基于所述新的映射后地址访问内存子系统。

[0107] 在本发明实施例中,映射方案配置寄存器还用于:配置有GPU应用对应的最佳访存映射方式。

[0108] 在本发明实施例中,所述GPU应用对应的最佳访存映射方式按照如下方式确定:

[0109] 获取GPU应用的GPU访问请求;

[0110] 利用不同的访存映射方式,将GPU访问请求进行不同的内存地址映射,获得不同映射后的GPU访问请求;

[0111] 确定不同映射后的GPU访问请求对应的信息熵;

[0112] 将最大信息熵对应的访存映射方式作为所述GPU应用的最佳访存映射方式。

[0113] 在本发明实施例中,确定不同映射后的GPU访问请求对应的信息熵,包括:

[0114] 根据GPU最大等待队列的长度确定一个时间段;

[0115] 基于所述时间段确定多个统计时间段;

[0116] 确定每个统计时间段内的GPU访问请求在不同访存通道上的分布信息熵;

[0117] 根据每个统计时间段内的分布信息熵确定多个统计时间段的分布信息熵平均值，将所述分布信息熵平均值作为映射后的GPU访问请求对应的信息熵。

[0118] 在本发明实施例中，按照如下方式配置GPU应用对应的最佳访存映射方式：

[0119] 将所述最佳访存映射方式对应的异或控制向量写入映射方案配置寄存器中，所述异或控制向量的长度与内存地址比特中已被选定进行异或的比特总数相同。

[0120] 在本发明实施例中，按照如下方式配置GPU应用对应的最佳访存映射方式：

[0121] 在将所述最佳访存映射方式对应的异或控制向量写入映射方案配置寄存器中的同时，将GPU应用对应的所述任务类型标识标记在内存管理单元的对应页表中，所述任务类型标识与异或控制向量存在对应关系。

[0122] 在本发明实施例中，如图9所示，还包括：

[0123] 片上性能寄存器，用于获取GPU应用的内存访问记录，根据所述GPU应用的内存访问记录更新GPU应用的任务类型标识。

[0124] 本发明实施例中，与现有技术中同时存在多种映射，或针对多种应用选取不同映射方式时，内存地址重叠会导致错误，导致现有技术无法扩展到对多种应用自适应选择方案的技术方案相比，本发明通过逻辑运算单元接收GPU应用发送的包括有任务类型信息的完成任务所需指令，根据所述完成任务所需指令执行相应操作，并在访问内存时向内存管理单元发起虚拟地址请求；内存管理单元根据所述虚拟地址请求，通过扩展页表将虚拟地址转换为物理地址，将物理地址发送至地址转换单元，基于任务类型信息查找到GPU应用对应的所述任务类型标识，将所述任务类型标识发送至映射方案配置寄存器，其中，所述任务类型标识标记在内存管理单元的扩展页表中；映射方案配置寄存器根据所述任务类型标识查找到对应的最佳访存映射方式，将所述最佳访存映射方式发送至地址转换单元；地址转换单元根据所述最佳访存映射方式，将所述物理地址映射为新的映射后地址，将所述新的映射后地址发送至内存子系统，基于所述新的映射后地址访问内存子系统。本发明可以获得如下有益效果：

[0125] 1、通过抓取GPU访存序列，并统计不同映射方案的信息熵，以用于量化评估内存地址映射方案的优劣，使得无需全系统仿真即可找到最佳的映射方式，能够为每个GPU应用提供定制化的内存伪随机交织映射方案，降低了设计空间探索复杂度，在针对性地提高GPU在特定任务下的性能的同时，保留GPU执行多种任务的能力。

[0126] 2、利用异或控制向量与片上可配置寄存器，实现了在保持硬件资源不变的情况下，内存地址映射方案的动态配置，提高了GPU在不同应用场景下的灵活性。

[0127] 3、通过软件可访问的寄存器配置通道，实现了在保持硬件资源不变的情况下，产品发布后通过软件进行内存映射方案的更新，提高了GPU在不同应用场景下的灵活性。

[0128] 4、支持通过片上性能计数器的统计信息，由软件收集后自动为应用分配适当的任务类型标识，以实现内存映射方案的自适应优化。

[0129] 5、通过片上地址转换单元和支持存储多种GPU访存映射方案的寄存器，实现可选择访存映射方式。

[0130] 6、通过扩展页表项内容，加入任务类型标记，使每个应用可获得定制化的内存映射方案，实现在GPU运行时分应用进行针对性的访存优化；

[0131] 7、通过利用页表对内存空间的划分，解决了多种访存映射方案共存时的冲突问

题,保证了访存的正确性。

[0132] 本发明具有较强泛用性,可应用于GPU内存层次结构中的任意具有并行访问能力的存储结构中(如片上缓存等)。事实上,以上方案尽管是针对物理内存地址的映射,但针对虚拟地址空间可采用同样的方式进行映射,从而能够从软件上实现灵活的地址映射方案。可应用于GPU内存层次结构中任意能够通过将一连串访存请求在若干个独立单元上平均分配时,能够获得性能增益的模块。例如缓存中,使请求在不同组之间平均分配以提高缓存利用率;在Dram中,将请求分散到不同的bank以提高总体访问带宽等等。

[0133] 特别的,如图10所示,对于在内存管理单元将虚拟地址映射到物理地址之前,根据包括但不限于进程ID(GPU内部分配给每个进程的唯一号码),虚拟地址范围(如0-4GB适用一种方案,4-8GB适用另一种方案),利用寄存器等方式进行进程ID与任务类型的配置,并利用与进程ID绑定的任务类型标记等等方式,来针对性地适用不同的地址转换策略,应属于本发明的在特定设计下的变型,也应收到本发明保护。

[0134] 本领域内的技术人员应明白,本发明的实施例可提供为方法、系统、或计算机程序产品。因此,本发明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,本发明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。

[0135] 本发明是参照根据本发明实施例的方法、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。

[0136] 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。

[0137] 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。

[0138] 以上所述的具体实施例,对本发明的目的、技术方案和有益效果进行了进一步详细说明,所应理解的是,以上所述仅为本发明的具体实施例而已,并不用于限定本发明的保护范围,凡在本发明的精神和原则之内,所做的任何修改、等同替换、改进等,均应包含在本发明的保护范围之内。

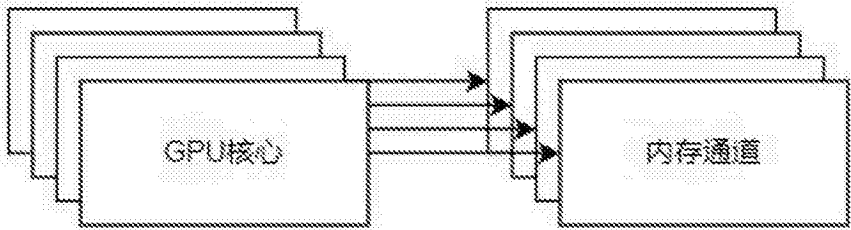


图1

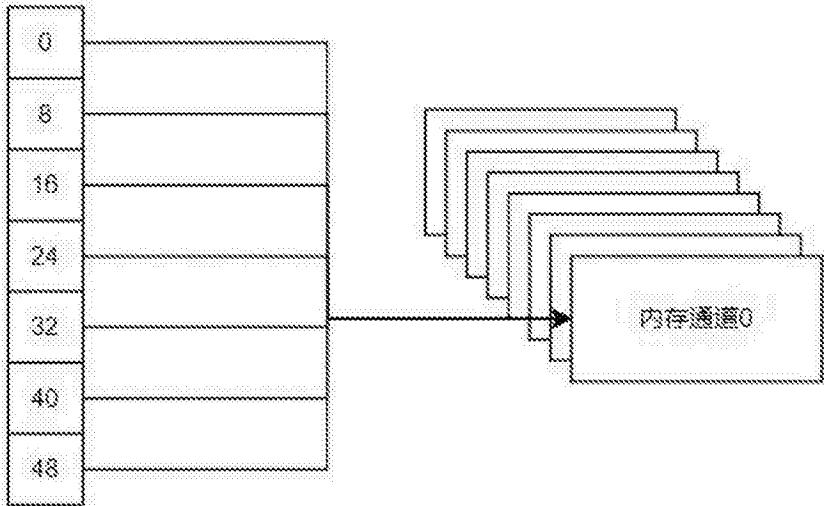


图2

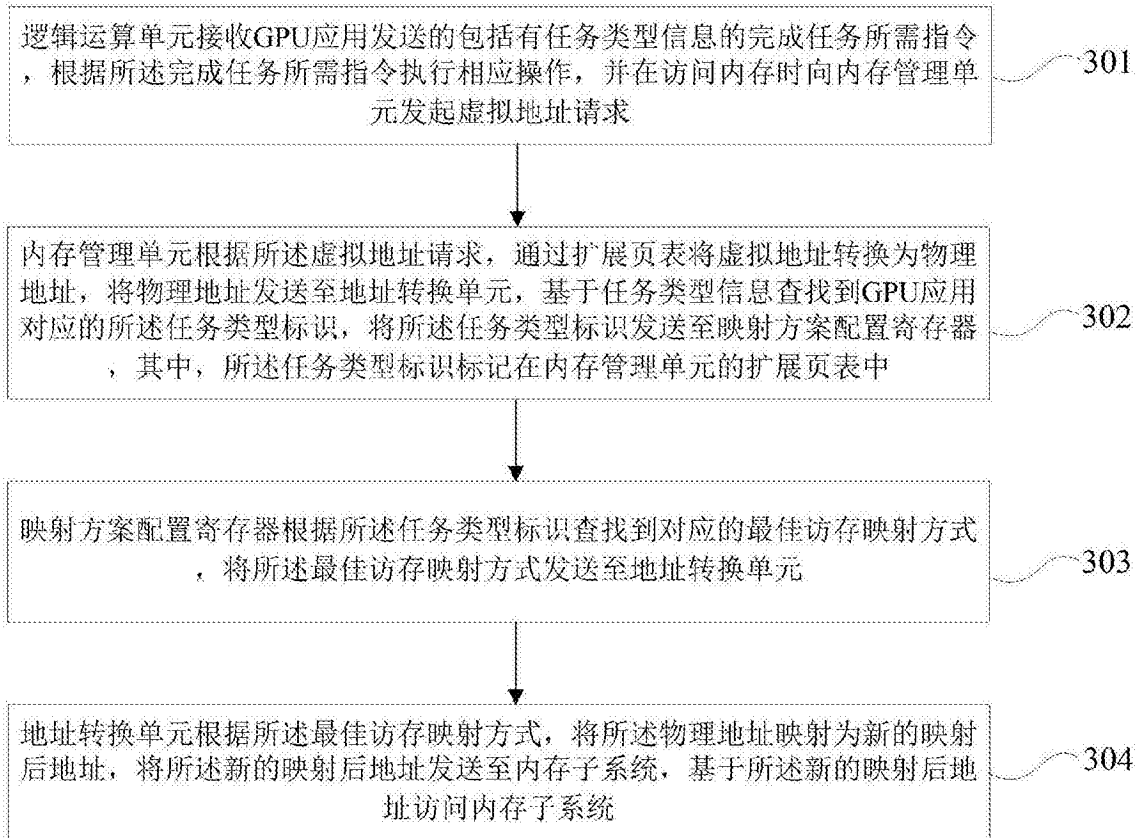


图3

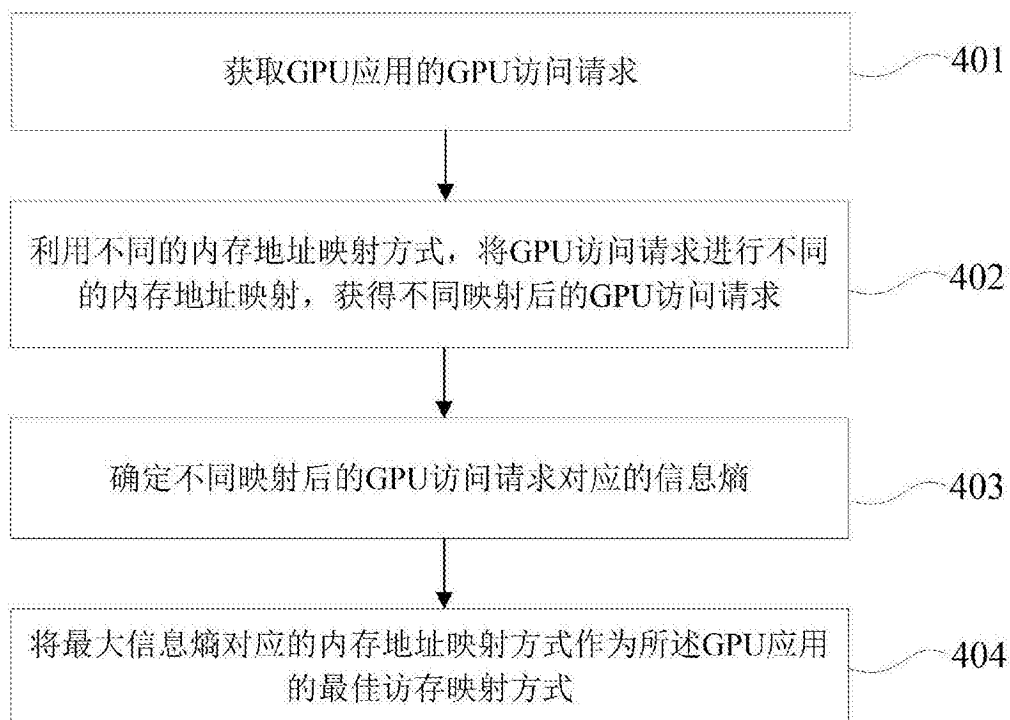


图4

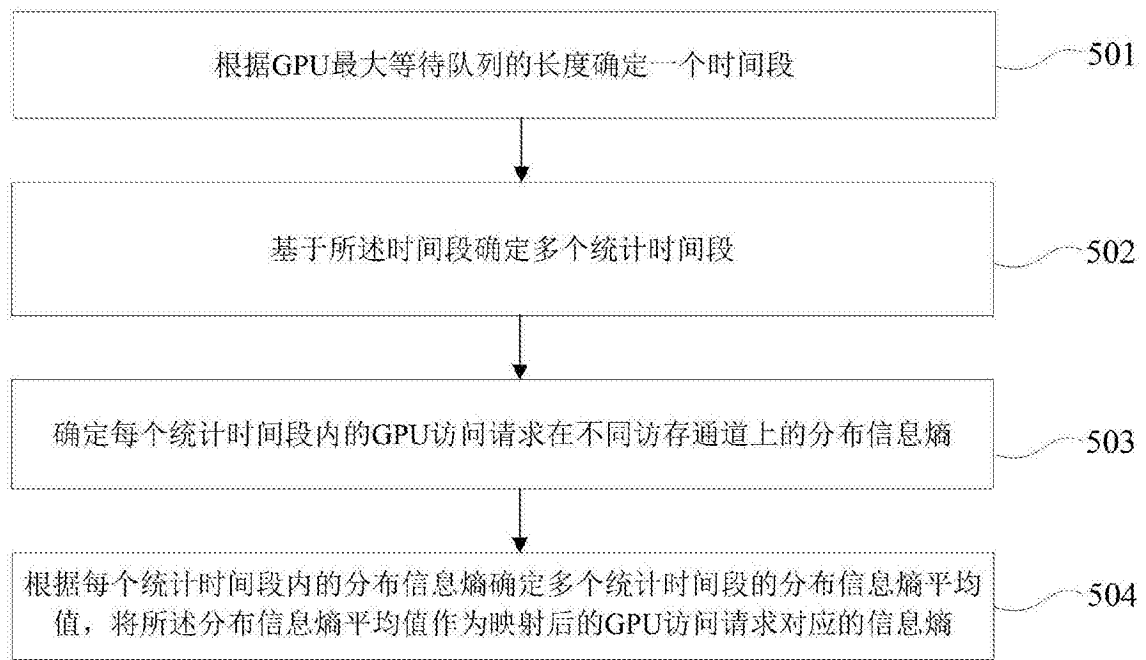


图5

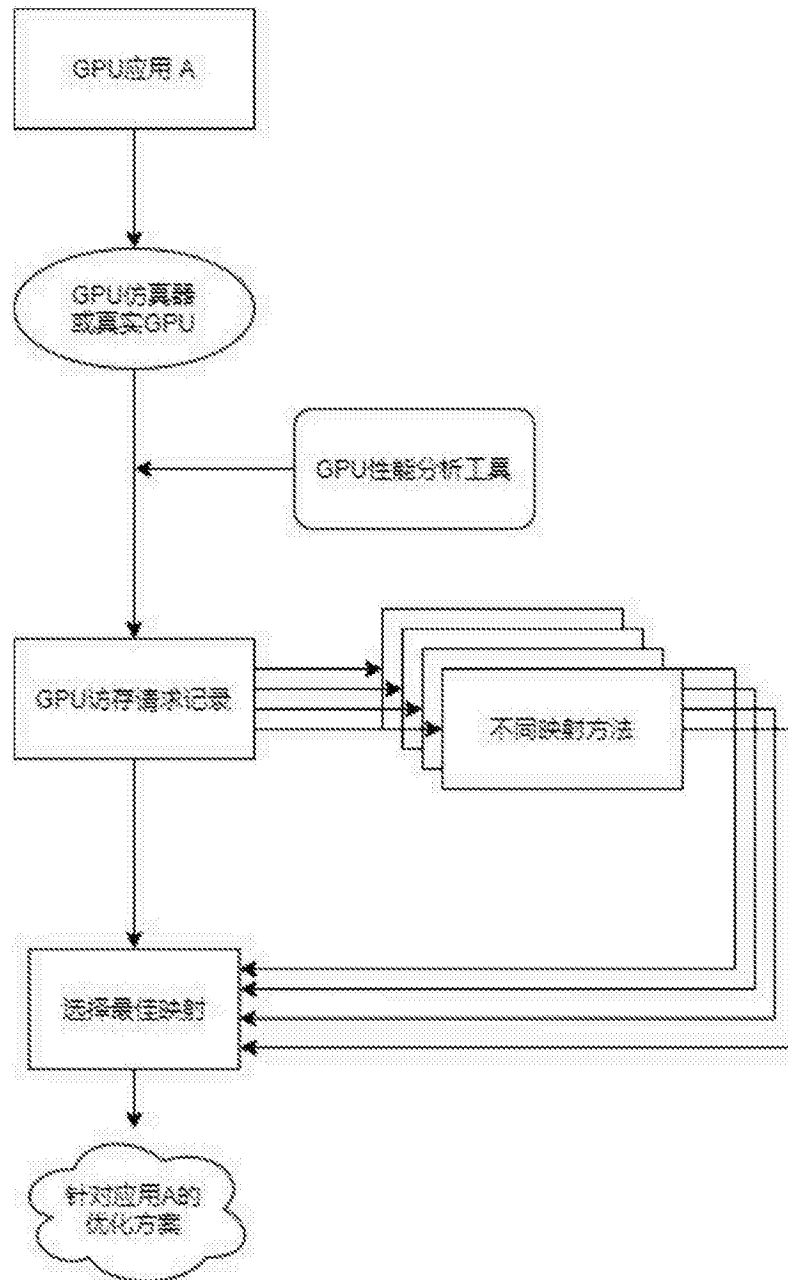


图6

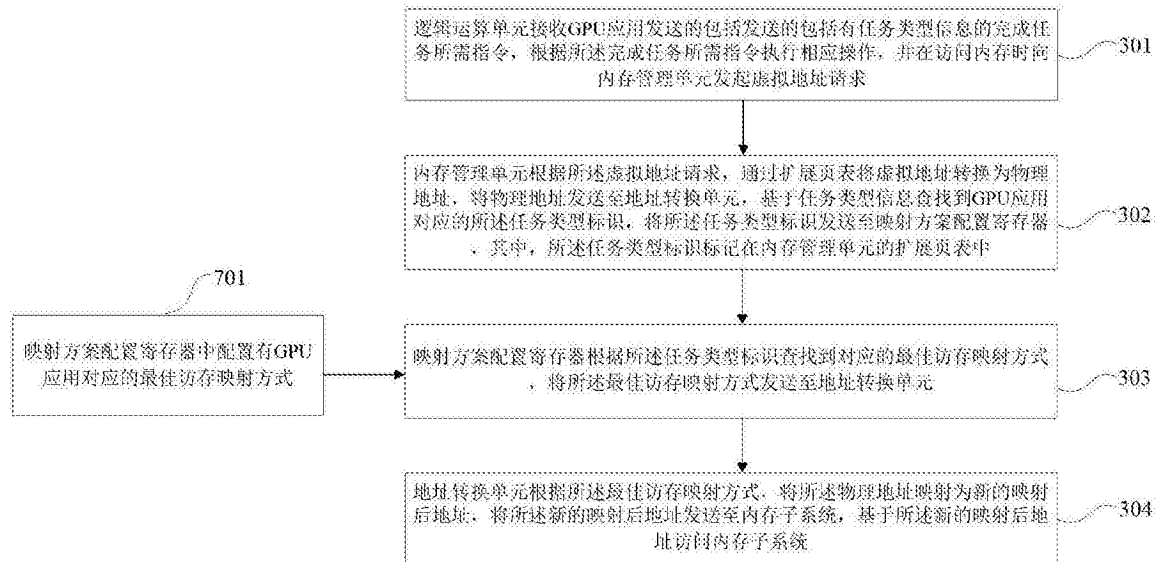


图7

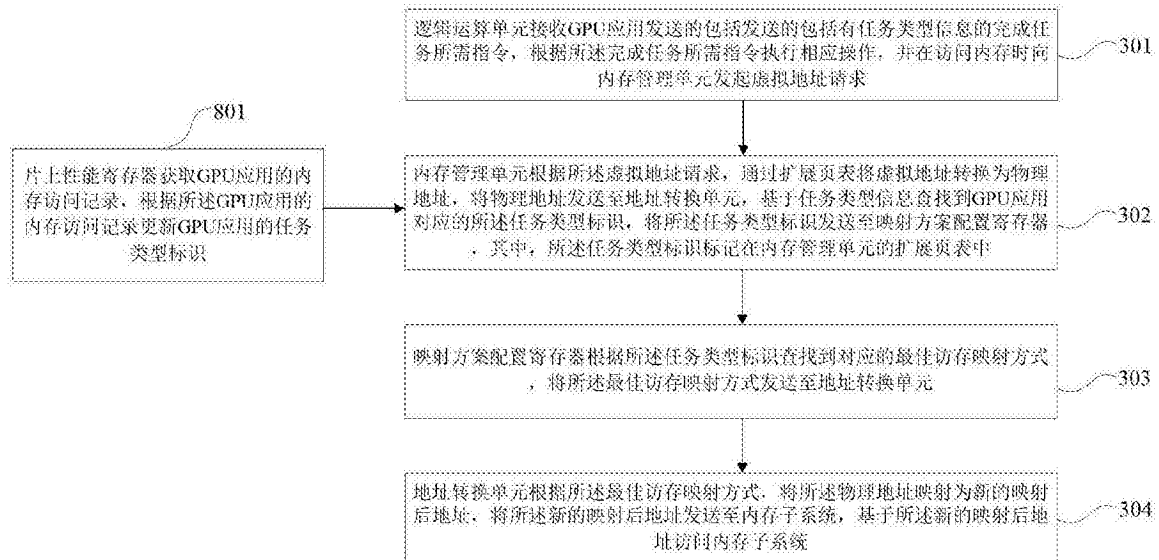


图8

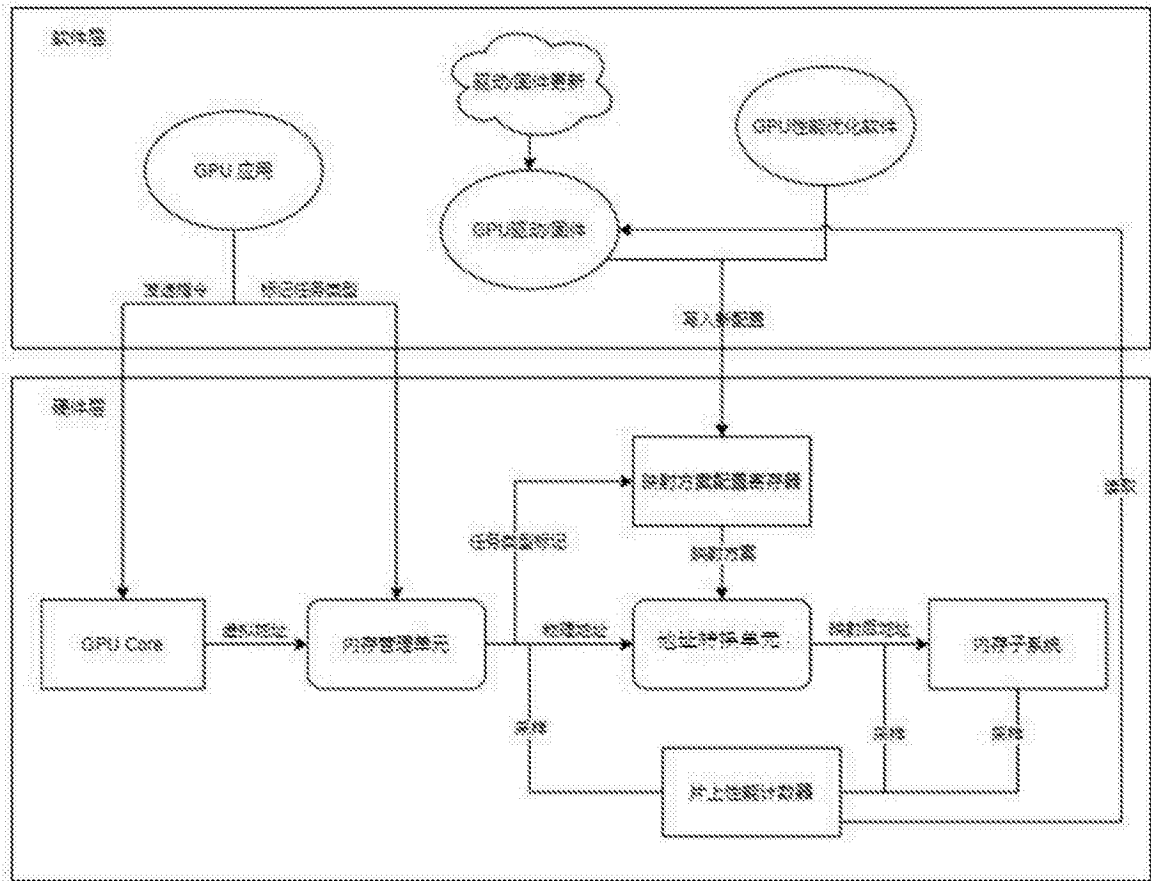


图9

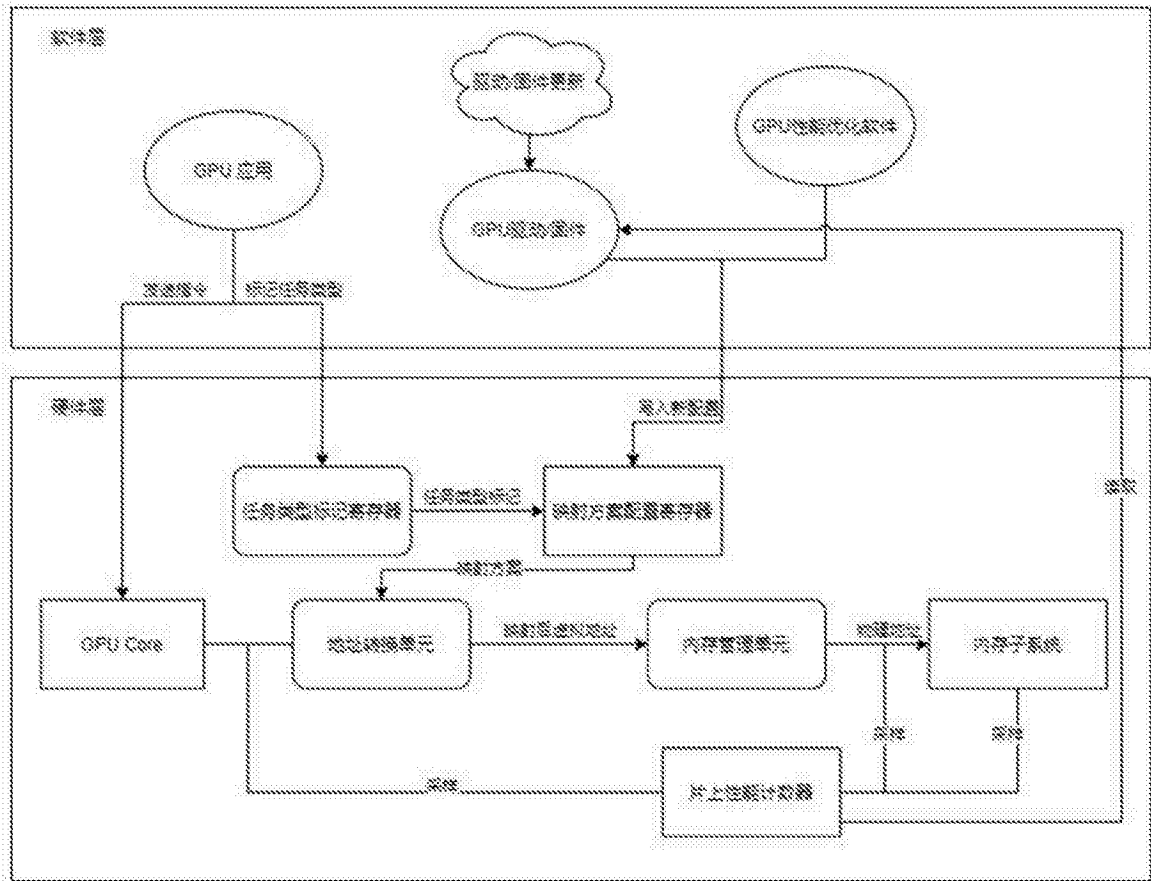


图10