(12) 按照专利合作条约所公布的国际申请

(19) 世界知识产权组织
国 际 局

(43) 国际公布日
2016 年 12 月 22 日 (22.12.2016)

WIPO | PCT

(10) 国际公布号
WO 2016/202154 A1

(51) 国际专利分类号:
G06F 9/50 (2006.01)

(21) 国际申请号: PCT/CN2016/083315

(22) 国际申请日: 2016 年 5 月 25 日 (25.05.2016)

(25) 申请语言: 中文

(26) 公布语言: 中文

(30) 优先权:
201510346334.9 2015 年 6 月 19 日 (19.06.2015) CN

(71) 申请人: **华为技术有限公司 (HUAWEI TECHNOLOGIES CO., LTD.)** [CN/CN]; 中国广东省深圳市龙岗区坂田华为总部办公楼, Guangdong 518129 (CN)。

(72) 发明人: **王聪 (WANG, Cong)**; 中国北京市海淀区中关村科学院南路 6 号, Beijing 100190 (CN)。 **展旭升 (ZHAN, Xusheng)**; 中国北京市海淀区中关村科学院南路 6 号, Beijing 100190 (CN)。 **包云岗 (BAO, Yungang)**; 中国北京市海淀区中关村科学院南路 6 号, Beijing 100190 (CN)。

(81) 指定国 (除另有指明，要求每一种可提供的国家保护): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW。

(84) 指定国 (除另有指明，要求每一种可提供的地区保护): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), 欧亚 (AM, AZ, BY, KG, KZ, RU, TJ, TM), 欧洲 (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG)。

根据细则 4.17 的声明:

— 关于申请人有权要求在先申请的优先权(细则 4.17(iii))

[见续页]

(54) **Title**: GPU RESOURCE ALLOCATION METHOD AND SYSTEM

(54) **发明名称** : 一种 GPU 资源的分配方法及系统

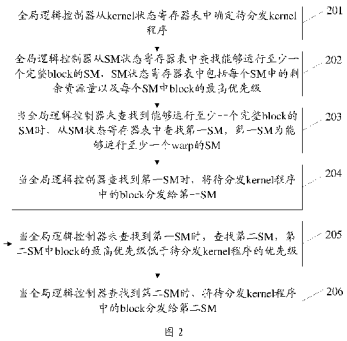

图 2

201 A global logic controller determining a kernel program to be distributed from a kernel state register table

202 The global logic controller looking up an SM capable of running at least one complete block from an SM state register table, the SM state register table comprising the remaining resource quantity in each SM and the highest priority of blocks in each SM

203 When the global logic controller does not find the SM capable of running at least one complete block, looking up a first SM from the SM state register table, the first SM being an SM capable of running at least one warp

204 When the global logic controller finds the first SM, distributing blocks in the kernel program to be distributed to the first SM

205 When the global logic controller does not find the first SM, looking up a second SM, wherein the highest priority of the blocks in the second SM is lower than the priority of the kernel program to be distributed

206 When the global logic controller finds the second SM, distributing the blocks in the kernel program to be distributed to the second SM

(57) **Abstract**: Disclosed are a GPU resource allocation method and system, which relate to the technical field of computers and can solve the problem that a kernel program with a high priority cannot be responded to in time. The method comprises: a global logic controller (1011) determining a kernel program to be distributed from a kernel state register table (1012); looking up an SM capable of running at least one complete thread block from an SM state register table (1013); when the SM capable of running at least one complete block is not found, looking up a first SM from the SM state register table (1013), the first SM being an SM capable of running at least one tread beam warp; when the first SM is found, distributing blocks in the kernel program to be distributed to the first SM; and when the first SM is not found, looking up a second SM, and then distributing the blocks in the kernel program to be distributed to the second SM. The present invention is applicable to GPU resource allocation.

(57) **摘要**:

[见续页]

WO 2016/202154 A1

本国际公布:

— 包括国际检索报告(条约第 21 条(3))。

一种 GPU 资源的分配方法及系统，涉及计算机技术领域，可以解决高优先级的 kernel 程序得不到及时响应的问题。通过全局逻辑控制器（1011）从核 kernel 状态寄存器表（1012）中确定待分发 kernel 程序；从 SM 状态寄存器表（1013）中查找能够运行至少一个完整线程块 block 的 SM；当未查找到能够运行至少一个完整 block 的 SM 时，从 SM 状态寄存器表（1013）中查找第一 SM，第一 SM 为能够运行至少一个线程束 warp 的 SM；当查找到第一 SM 时，将待分发 kernel 程序中的 block 分发给第一 SM；当未查找到第一 SM 时，查找第二 SM，进而将待分发 kernel 程序中的 block 分发给第二 SM。其适于 GPU 资源分配时采用。

一种 GPU 资源的分配方法及系统

技术领域

本发明涉及计算机技术领域，尤其涉及一种 GPU 资源的分配方法及系统。

背景技术

随着 GPU（Graphic Processing Unit，图形处理器）通用技术的发展，GPU 不仅能够处理图像负载，也能够处理特定类型的通用程序。目前，当有多个不同的 kernel(核)程序需要访问 GPU 时，一般是以序列化的方式使请求访问 GPU 的 kernel 程序按照发送请求的时间顺序逐个访问 GPU。如果一个延迟很长的 kernel 程序正在占用 GPU，当有优先级更高的 kernel 程序需要访问 GPU 时，必须等前面正在访问 GPU 的 kernel 程序以及正在等待访问 GPU 的 kernel 程序运行结束后，释放出 GPU 中的 SM（Stream Multiprocessor，流式多处理器）资源，该优先级更高的 kernel 程序才能访问 GPU，使得该优先级更高的 kernel 程序得不到及时响应，影响业务质量。

为了避免延时长的 kernel 程序长时间独占 GPU 中的 SM 资源，当有高优先级的 kernel 程序需要访问 GPU 时，可以查找空闲的 SM，当查找到空闲的 SM 时，将高优先级的 kernel 程序分发给该空闲的 SM 运行。

然而，如果 GPU 中没有空闲的 SM，则需要等待 GPU 中出现空闲的 SM 时，才能够开始运行高优先级的 kernel 程序，导致高优先级的 kernel 程序得不到及时的响应。

发明内容

本发明的实施例提供一种 GPU 资源的分配方法及系统，可以解决高优先级的 kernel 程序得不到及时响应的问题。

为达到上述目的，本发明的实施例采用如下技术方案：

第一方面，本发明实施例提供一种图形处理器 GPU 资源的分配方法，所述方法应用于 GPU 资源的分配系统中，所述系统包括全局逻辑控制器以及至少两个能够与所述全局逻辑控制器通信的流式多处理器 SM，所述方法包括：

所述全局逻辑控制器从核 kernel 状态寄存器表中确定待分发 kernel 程序，所述 kernel 状态寄存器表中包括每个未完成运行的 kernel 程序的优先级以及每

个未完成运行的 kernel 程序中未分发的线程块 block 数量，所述待分发 kernel 程序为所述 kernel 状态寄存器表中优先级最高且未分发的 block 数量不为零的 kernel 程序;

所述全局逻辑控制器从 SM 状态寄存器表中查找能够运行至少一个完整 block 的 SM，所述 SM 状态寄存器表中包括每个 SM 中的剩余资源量以及每个 SM 中 block 的最高优先级;

当所述全局逻辑控制器未查找到能够运行至少一个完整 block 的 SM 时，从所述 SM 状态寄存器表中查找第一 SM，所述第一 SM 为能够运行至少一个线程束 warp 的 SM;

当所述全局逻辑控制器查找到所述第一 SM 时，将所述待分发 kernel 程序中的 block 分发给所述第一 SM;

当所述全局逻辑控制器未查找到所述第一 SM 时，查找第二 SM，所述第二 SM 中 block 的最高优先级低于所述待分发 kernel 程序的优先级;

当所述全局逻辑控制器查找到所述第二 SM 时，将所述待分发 kernel 程序中的 block 分发给所述第二 SM。

在第一种可能的实施例中，结合第一方面，在所述全局逻辑控制器从 SM 状态寄存器表中查找能够运行至少一个完整 block 的 SM 之后，所述方法还包括:

当所述全局逻辑控制器查找到能够运行至少一个完整 block 的 SM 时，确定第一数量，所述第一数量为所述能够运行一个完整 block 的 SM 实际能够运行的 block 的数量;

当所述待分发 kernel 程序中未分发的 block 的数量大于所述第一数量时，将所述待分发 kernel 程序中所述第一数量的 block 分发给所述能够运行至少一个完整 block 的 SM;

当所述待分发 kernel 程序中未分发的 block 的数量小于或等于所述第一数量时，将所述待分发 kernel 程序中未分发的 block 全部分发给所述能够运行至少一个完整 block 的 SM。

在第二种可能的实施例中，结合第一方面中的第一种可能的实施例，在所述全局逻辑控制器将所述待分发 kernel 程序中的 block 分发给所述第二 SM 之后，所述方法还包括:

第二 SM 逻辑控制器从 block 状态寄存器表中确定优先级最高的 block，所

述第二 SM 逻辑控制器为所述第二 SM 中的 SM 逻辑控制器，所述 block 状态寄存器表包括被分发到所述第二 SM 中的每个 block 的优先级;

所述第二 SM 逻辑控制器查找当前的空闲硬件 warp;

当所述第二 SM 逻辑控制器确定所述空闲硬件 warp 能够运行一个 warp，且未接收到优先级更高的 block 时，将所述优先级最高的 block 中的一个 warp 分发给所述空闲硬件 warp。

在第三种可能的实施例中，结合第一方面或第一方面中上述任一种可能的实施例，所述 SM 状态寄存器表中包括每个 SM 的剩余寄存器数量、剩余硬件 warp 数量以及剩余共享存储空间，所述第一 SM 为所述剩余寄存器数量大于运行一个 warp 所需的寄存器数量、所述剩余硬件 warp 数量大于运行一个 warp 所需的硬件 warp 数量且所述剩余共享存储空间大于运行一个 warp 所需的共享存储空间的 SM。

在第四种可能的实施例中，结合第一方面中第三种可能的实施例，在所述当所述第二 SM 逻辑控制器确定所述空闲硬件 warp 能够运行一个 warp，且未接收到优先级更高的 block 时，将所述优先级最高的 block 中的一个 warp 分发给所述硬件 warp 之后，所述方法还包括:

所述第二 SM 逻辑控制器确定所述第二 SM 中有运行完成的 warp 时，通知所述全局逻辑控制器更新所述第二 SM 的剩余寄存器数量、剩余 warp 数量以及剩余共享存储空间;

当所述第二 SM 逻辑控制器确定所述运行完成的 warp 所属 block 中不存在未运行的 warp 时，确定所述第二 SM 中未运行完成的 block 的最高优先级，通知所述全局逻辑控制器更新所述 SM 状态寄存器表中的所述第二 SM 中 block 的最高优先级。

第二方面，本发明实施例提供一种图形处理器 GPU 资源的分配系统，所述系统包括全局逻辑控制器以及至少两个能够与所述全局逻辑控制器通信的流式多处理器 SM; 所述全局逻辑控制器包括: 第一确定单元、第一查找单元以及第一分发单元;

所述第一确定单元，用于从核 kernel 状态寄存器表中确定待分发 kernel 程序，所述 kernel 状态寄存器表中包括每个未完成运行的 kernel 程序的优先级以及每个未完成运行的kernel程序中未分发的线程块block数量，所述待分发kernel

程序为所述 kernel 状态寄存器表中优先级最高且未分发的 block 数量不为零的 kernel 程序;

所述第一查找单元，用于从 SM 状态寄存器表中查找能够运行至少一个完整 block 的 SM，所述 SM 状态寄存器表中包括每个 SM 中的剩余资源量以及每个 SM 中 block 的最高优先级；当未查找到能够运行至少一个完整 block 的 SM 时，从所述 SM 状态寄存器表中查找第一 SM，所述第一 SM 为能够运行至少一个线程束 warp 的 SM;

所述第一分发单元，用于当所述第一查找单元查找到所述第一 SM 时，将所述待分发 kernel 程序中的 block 分发给所述第一 SM;

所述第一 SM，用于运行所述第一分发单元分发的所述待分发 kernel 程序中的 block;

所述第一查找单元，还用于当未查找到所述第一 SM 时，查找第二 SM，所述第二 SM 中 block 的最高优先级低于所述待分发 kernel 程序的优先级;

所述第一分发单元，还用于当所述第一查找单元查找到所述第二 SM 时，将所述待分发 kernel 程序中的 block 分发给所述第二 SM;

所述第二 SM，用于运行所述第一分发单元分发的所述待分发 kernel 程序中的 block。

在第一种可能的实施例中，结合第二方面，所述第一确定单元，还用于当所述第一查找单元查找到能够运行至少一个完整 block 的 SM 时，确定第一数量，所述第一数量为所述能够运行一个完整 block 的 SM 实际能够运行的 block 的数量;

所述第一分发单元，还用于当所述待分发 kernel 程序中未分发的 block 的数量大于所述第一数量时，将所述待分发 kernel 程序中所述第一数量的 block 分发给所述能够运行至少一个完整 block 的 SM；当所述待分发 kernel 程序中未分发的 block 的数量小于或等于所述第一数量时，将所述待分发 kernel 程序中未分发的 block 全部分发给所述能够运行至少一个完整 block 的 SM;

所述能够运行至少一个完整 block 的 SM，用于运行所述第一分发单元分发的所述待分发 kernel 程序中的 block。

在第二种可能的实施例中，结合第一方面中的第一种可能的实施例，所述第二 SM 包括第二确定单元、第二查找单元以及第二分发单元;

所述第二确定单元，用于从block状态寄存器表中确定优先级最高的block，所述block状态寄存器表包括被分发到所述第二SM中的每个block的优先级；

所述第二查找单元，用于查找当前的空闲硬件warp；

所述第二分发单元，用于确定所述空闲硬件warp能够运行一个warp，且未接收到优先级更高的block时，将所述优先级最高的block中的一个warp分发给所述空闲硬件warp。

在第三种可能的实施例中，结合第二方面或第二方面中上述任一种可能的实施例，所述SM状态寄存器表中包括每个SM的剩余寄存器数量、剩余硬件warp数量以及剩余共享存储空间，所述第一SM为所述剩余寄存器数量大于运行一个warp所需的寄存器数量、所述剩余硬件warp数量大于运行一个warp所需的硬件warp数量且所述剩余共享存储空间大于运行一个warp所需的共享存储空间的SM。

在第四种可能的实施例中，结合第二方面中的第三种可能的实施例，所述第二SM中还包括通知单元；

所述通知单元，用于当确定所述第二SM中有运行完成的warp时，通知所述全局逻辑控制器更新所述第二SM的剩余寄存器数量、剩余warp数量以及剩余共享存储空间；当所确定所述运行完成的warp所属block中不存在未运行的warp时，确定所述第二SM中未运行完成的block的最高优先级，通知所述全局逻辑控制器更新所述SM状态寄存器表中的所述第二SM中block的最高优先级。

本发明实施例提供的GPU资源的分配方法及系统，全局逻辑控制器从kernel状态寄存器表中确定待分发kernel程序，从SM状态寄存器表中查找能够运行至少一个完整block的SM，当未查找到能够运行至少一个block的SM时，则继续查找能够运行至少一个warp的第一SM，将待分发kernel程序中的一个block分发给第一SM，当未查找到第一SM时，将待分发kernel程序中的block分发给第二SM。与现有技术中必须等待GPU中有空闲的SM时，才能将高优先级kernel中的block分发给该SM而导致高优先级的kernel程序得不到及时响应相比，本发明实施例中，当未查找到能够运行至少一个block的SM时，不是等待其他kernel程序释放资源，而是查找能够运行至少一个warp的第一SM，由于warp比block小，所以运行完一个warp比运行完一个block更快，所以更容易查找到能够运行至少一个warp的SM，查找到之后就可以将待分发kernel程序

的一个block分发给第一SM，无需等待低优先级的kernel程序运行完一个block，当未查找到能够运行至少一个 warp 的第一 SM 时，将待分发 kernel 程序中的block 分发给第二 SM，减少了分发 block 的等待时间，提高了高优先级的 kernel 程序的响应速度。

附图说明

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。

图 1 为本发明实施例提供的一种 GPU 资源的分配系统的逻辑结构示意图；

图 2 为本发明实施例提供的一种 GPU 资源的分配方法的流程图；

图 3 为本发明实施例提供的另一种 GPU 资源的分配方法的流程图；

图 4 为本发明实施例提供的另一种 GPU 资源的分配方法的流程图；

图 5 为本发明实施例提供的另一种 GPU 资源的分配方法的流程图；

图 6 为本发明实施例提供的另一种 GPU 资源的分配系统的逻辑结构示意图；

图 7 为本发明实施例提供的另一种 GPU 资源的分配系统的逻辑结构示意图；

图 8 为本发明实施例提供的一种 GPU 资源的分配装置的逻辑结构示意图；

图 9 为本发明实施例提供的一种 GPU 资源的分配装置的逻辑结构示意图。

具体实施方式

下面将结合本发明实施例中的附图，对本发明实施例中的技术方案进行清楚、完整地描述，显然，所描述的实施例仅仅是本发明一部分实施例，而不是全部的实施例。基于本发明中的实施例，本领域普通技术人员在没有作出创造性劳动前提下所获得的所有其他实施例，都属于本发明保护的范围。

本发明实施应用于 GPU 资源的分配系统中，如图 1 所示，该系统包括全局调度器 101 以及能够与全局调度器 101 通信的至少两个 SM 102。

其中，全局调度器 101 包括：全局逻辑控制器 1011、kernel 状态寄存器表 1012 以及 SM 状态寄存器表 1013。

SM102 包括: SM 逻辑控制器 1021 以及 block(线程块)状态寄存器表 1022。

全局调度器 101，用于将 kernel 程序分发给 SM102 运行。

全局逻辑控制器 1011，用于根据 kernel 状态寄存器表 1012 以及 SM 状态寄存器表 1013 将 kernel 程序以 block 为粒度或者以 warp（线程束）为粒度分发给 SM102。

需要说明的是，在本发明实施例中 kernel（核）程序为能够在 GPU 上运行的程序，一个 kernel 程序包含至少两个 block（线程块），一个 block 包含至少两个 warp（线程束），warp 为至少两个 GPU 线程组成的一组线程，一般情况下，一个 warp 由 32 个 GPU 线程组成。

kernel 状态寄存器表 1012，用于存储每个未完成运行的 kernel 程序信息。

其中，kernel 程序信息包括 kernel 程序的优先级、运行该 kernel 程序需要的寄存器数量、运行该 kernel 程序需要的共享存储空间、该 kernel 程序中还未分发的 block 数量。

SM 状态寄存器表 1013,用于存储每个 SM102 当前的剩余资源量以及每个 SM 中 block 的最高优先级。

其中，每个 SM102 当前的剩余资源量包括剩余的寄存器数量、剩余的硬件 warp 数量以及剩余的共享存储空间。

SM102，用于运行全局调度器 101 分发的 kernel 程序。

SM 逻辑控制器 1021,用于根据 block 状态寄存器表将 block 中的 warp 分发给硬件 warp 运行。

block 状态寄存器表 1022，用于存储每个 block 的运行情况。

其中,block 的运行情况包括 block 的优先级、block 所属 kernel 的编号,block 在 kernel 中的编号,block 中未运行部分所需的寄存器数和所需的共享存储空间,以及 block 中还未分发的 warp 数。

为了加快高优先级的 kernel 程序的响应速度，本发明实施例提供一种 GPU 资源的分配方法,该方法应用于图 1 所示的 GPU 资源分配系统中,如图 2 所示,该方法包括:

201、全局逻辑控制器从 kernel 状态寄存器表中确定待分发 kernel 程序。

其中,kernel 状态寄存器表中包括每个未完成运行的 kernel 程序的优先级以及每个未完成运行的 kernel 程序中未分发的 block 数量，待分发 kernel 程序为 kernel 状态寄存器表中优先级最高且未分发的 block 数量不为零的 kernel 程序。

202、全局逻辑控制器从 SM 状态寄存器表中查找能够运行至少一个完整

block 的 SM，SM 状态寄存器表中包括每个 SM 中的剩余资源量以及每个 SM 中 block 的最高优先级。

其中，SM 状态寄存器表中具体包括每个 SM 的剩余寄存器数量、剩余硬件 warp 数量以及剩余共享存储空间以及每个 SM 中 block 的最高优先级，block 的优先级与 block 所属 kernel 的优先级相同。

需要说明的是，一个 SM 中可能会存在多个未运行完成的 block，且这些 block 属于相同或者不同的 kernel 程序，全局逻辑控制器为了合理的分发 block，需确定同一 SM 中每个 block 的优先级，从而确定这些 block 中最高的优先级，再将该 SM 中 block 最高优先级存储在 SM 状态寄存器表中。

能够运行至少一个完整 block 的 SM 为剩余寄存器数量大于运行一个 block 所需的寄存器数量、剩余硬件 warp 数量大于运行一个 block 所需的硬件 warp 数量且剩余共享存储空间大于运行一个 block 所需的共享存储空间的 SM。

举例说明，例如运行一个 block 需要 36kb 的寄存器，而一个 SM 中只剩下 20kb 的寄存器，则该 SM 不能运行一个 block。

203、当全局逻辑控制器未查找到能够运行至少一个完整 block 的 SM 时，从 SM 状态寄存器表中查找第一 SM，第一 SM 为能够运行至少一个 warp 的 SM。

可以理解的是，第一 SM 为剩余寄存器数量大于运行一个 warp 所需的寄存器数量、剩余硬件 warp 数量大于运行一个 warp 所需的硬件 warp 数量且剩余共享存储空间大于运行一个 warp 所需的共享存储空间的 SM。

需要说明的是，当运行一个 block 需要 36kb 寄存器，而剩余资源量最多的 SM 只剩余 12kb 的寄存器时，全局逻辑控制器查找不到能够运行至少一个 block 的 SM，而运行一个 warp 只需要 6kb 寄存器，此时剩余 12kb 寄存器的 SM 可以运行两个 warp，即全局逻辑控制器能够查找到第一 SM。

204、当全局逻辑控制器查找到第一 SM 时，将待分发 kernel 程序中的 block 分发给第一 SM。

其中，如果第一 SM 的剩余资源只能运行一个 warp，则将待分发 kernel 程序中的 block 分发给第一 SM 后，第一 SM 会将 block 中的 warp 逐个运行。

205、当全局逻辑控制器未查找到第一 SM 时，查找第二 SM，第二 SM 中 block 的最高优先级低于待分发 kernel 程序的优先级。

206、当全局逻辑控制器查找到第二 SM 时，将待分发 kernel 程序中的 block

分发给第二 SM。

值得说明的是，为了避免当前确定的待分发 kernel 程序中的 block 抢占 SM 中正在运行的优先级更高的 kernel 程序所占用的资源，该优先级更高的 kernel 程序未分发的 block 数为 0，但是已分发的 block 还未运行结束，当全局逻辑控制器未查找到能够运行一个 warp 的 SM 时，需将 SM 状态寄存器表中存储的每个 SM 中 block 的最高优先级与步骤 201 中确定的待分发 kernel 程序的优先级相比较，将 SM 中 block 的最高优先级低于该待分发 kernel 程序的优先级的 SM 确定为第二 SM，再将待分发 kernel 程序中的 block 分发给第二 SM。

此外，需要说明的是，第二 SM 的数量为至少一个，在第二 SM 中的剩余资源不足以运行一个 warp 时，也将待分发 kernel 程序中的 block 分发给第二 SM，可以减少分发 block 等待的时间，将待分发 kernel 程序中的 block 分发给第二 SM 之后，第二 SM 中有 warp 运行结束就可以开始运行该待分发 kernel 程序中的 block。

本发明实施例提供的 GPU 资源的分配方法，全局逻辑控制器从 kernel 状态寄存器表中确定待分发 kernel 程序，从 SM 状态寄存器表中查找能够运行至少一个完整 block 的 SM，当未查找到能够运行至少一个 block 的 SM 时，则继续查找能够运行至少一个 warp 的第一 SM，将待分发 kernel 程序中的一个 block 分发给第一 SM，当未查找到第一 SM 时，将待分发 kernel 程序中的 block 分发给第二 SM。与现有技术中必须等待 GPU 中有空闲的 SM 时，才能将高优先级 kernel 中的 block 分发给该 SM 而导致高优先级的 kernel 程序得不到及时响应相比，本发明实施例中，当未查找到能够运行至少一个 block 的 SM 时，不是等待其他 kernel 程序释放资源，而是查找能够运行至少一个 warp 的第一 SM，由于 warp 比 block 小，所以运行完一个 warp 比运行完一个 block 更快，所以更容易查找到能够运行至少一个 warp 的 SM，查找到之后就可以将待分发 kernel 程序的一个 block 分发给第一 SM，无需等待低优先级的 kernel 程序运行完一个 block，当未查找到能够运行至少一个 warp 的第一 SM 时，将待分发 kernel 程序中的 block 分发给第二 SM，减少了分发 block 的等待时间，提高了高优先级的 kernel 程序的响应速度。

作为对上述实施例的补充，在本发明实施例提供的另一种实现方式中，如图 3 所示，在上述步骤 202、全局逻辑控制器从 SM 状态寄存器表中查找能够运

行至少一个完整 block 的 SM 之后，如果查找到能够运行至少一个完整 block 的 SM，则执行下述步骤 207 至 209。

207、当全局逻辑控制器查找到能够运行至少一个完整 block 的 SM 时，确定第一数量，第一数量为能够运行一个完整 block 的 SM 实际能够运行的 block 的数量。

其中，第一数量为全局逻辑控制器通过能够运行至少一个完整 block 的 SM 中的 SM 状态寄存器表确定的。全局逻辑控制器能够根据 SM 状态寄存器表中存储的 SM 的剩余资源量以及运行一个 block 所需的资源量计算出该 SM 实际能够运行的 block 数量。

208、当待分发 kernel 程序中未分发的 block 的数量大于第一数量时，将待分发 kernel 程序中第一数量的 block 分发给能够运行至少一个完整 block 的 SM。

其中，kernel 状态寄存器表中还包括每个未完成运行的 kernel 中未完成分发的 block 数量。

值得说明的是，当待分发 kernel 程序中未分发的 block 数量大于第一数量时，说明查找到的 SM 的剩余资源不足以运行待分发 kernel 程序中的未分发的 block，所以先将第一数量的 block 分发给该 SM，当有 block 运行完成释放出 SM 中的资源后，再将该 kernel 中剩余的 block 分发给 SM。

209、当待分发 kernel 程序中未分发的 block 的数量小于或等于第一数量时，将待分发 kernel 程序中的未分发的 block 全部分发给能够运行至少一个完整 block 的 SM。

值得说明的是，在上述步骤 204、206、208 以及 209 向全局逻辑控制器向 SM 分发 block 后，都需要更新 kernel 状态寄存器中待分发 kernel 程序中未分发的 block 数量。

本发明实施例提供的 GPU 资源的分配方法，当全局逻辑控制器查找到能够运行至少一个 block 的 SM 时，确定第一数量，当待分发 kernel 程序中未分发的 block 的数量大于第一数量时，将待分发 kernel 程序中第一数量的 block 分发给能够运行至少一个完整 block 的 SM；当待分发 kernel 程序中未分发的 block 的数量小于或等于第一数量时，将待分发 kernel 程序中的全部 block 分发给能够运行至少一个完整 block 的 SM。在能够查找到运行至少一个 block 的 SM 时，将待分发 kernel 中尽可能多的 block 分发给该 SM，可以使待分发 kernel 得到及时

的响应，提高了高优先级的 kernel 程序的响应速度。

在全局逻辑控制器将 block 分发给 SM 后，SM 需合理的将 block 中的 warp 分发运行，所以本发明另一实施例提供了在步骤 204、当全局逻辑控制器查找到第二 SM 时，将待分发 kernel 程序中的一个 block 分发给第二 SM 之后，第二 SM 逻辑控制器分发 warp 的方法，如图 4 所示，该方法包括:

401、第二 SM 逻辑控制器从 block 状态寄存器表中确定优先级最高的 block，第二 SM 逻辑控制器为第二 SM 中的 SM 逻辑控制器，block 状态寄存器表包括被分发到第二 SM 中的每个 block 的优先级。

结合图 1 所示的 GPU 资源的分配系统，全局逻辑控制器连接于至少两个 SM，当全局逻辑控制器将待分发 kernel 程序中的一个 block 分发给第二 SM 后，第二 SM 中的第二 SM 逻辑控制器需将该 block 中的 warp 分发给硬件 warp 运行。

由于第二 SM 中还正在运行其他 kernel 中的 block，或者还有其他 kernel 中的 block 正在等待运行，所以第二 SM 逻辑控制器需要从 block 状态寄存器表中确定优先级最高的 block，优先运行优先级最高的 block 中的 warp。

值得说明的是，block 状态寄存器中存储的 block 的优先级为 block 所属 kernel 的优先级，同一 kernel 中的 block 的优先级是相同的。

402、第二 SM 逻辑控制器查找当前的空闲硬件 warp。

需要说明的是，当第二 SM 逻辑控制器查找到空闲硬件 warp 时，则执行下述步骤 403；当第二 SM 逻辑控制器未查找到空闲硬件 warp 时，则重复查找动作，直到查找到空闲的硬件 warp 再继续执行下述步骤 403。

由于第二 SM 中还有低优先级的 kernel 程序正在运行，所以，等待低优先级的 kernel 程序中有 warp 运行结束后，就会有硬件 warp 恢复空闲状态，此时第二 SM 逻辑控制器就能够查找到空闲硬件 warp，高优先级的 kernel 程序中的 warp 即可占用该硬件 warp。

403、当第二 SM 逻辑控制器确定空闲硬件 warp 能够运行一个 warp，且未接收到优先级更高的 block 时，将优先级最高的 block 中的一个 warp 分发给空闲硬件 warp，并更新 block 状态寄存器表。

其中，判断空闲硬件 warp 是否能够运行一个 warp 的方法为：判断第二 SM 中的寄存器数量是否足够运行一个 warp，如果足够，且此时第二 SM 未接收到优先级更高的 block 时，则将此时优先级最高的 block 中的一个 warp 分发给查

找到的空闲硬件 warp；如果不够，则继续等待，直到有 warp 运行结束，寄存器数量足够运行一个 warp 时，再向该空闲硬件 warp 分发一个 warp。

值得说明的是，将优先级最高的 block 中的一个 warp 分发给空闲硬件 warp 后，还需判断该优先级最高的 block 是否分发完毕，若是，则重新执行上述步骤 401 至 403；若否，则重新执行上述步骤 402 至 403。

值得说明的是，在上述步骤 205、当全局逻辑控制器查找到第一 SM 时，将待分发 kernel 程序中的 block 分发给第一 SM 之后，第一 SM 的 SM 逻辑控制器分发 warp 的方法与第二 SM 逻辑控制器分发 warp 的方法相同，此处不再赘述。

本发明实施例提供的 GPU 资源的分配方法，第二 SM 逻辑控制器首先查找空闲硬件 warp，当查找到空闲硬件 warp 且此时第二 SM 能够运行一个 warp 时，就将优先级最高的 block 中的一个 warp 分发给硬件 warp 运行，无需等待第二 SM 中有能够运行整个 block 的资源后再将整个 block 分发给硬件 warp 运行，减少了等待时间，提高了高优先级 kernel 程序的响应速度。

为了减少 SM 中的空闲资源，提高 SM 的资源利用率，以加快高优先级 kernel 程序的响应速度，在本发明实施例提供的另一种实现方式中，如图 5 所示，该方法还包括:

501、当第二 SM 逻辑控制器确定第二 SM 中有运行完成的 warp 时，通知全局逻辑控制器更新 SM 状态寄存器表中的第二 SM 的剩余寄存器数量、剩余硬件 warp 数量以及剩余共享存储空间。

可以理解的是，当运行完一个 warp 时，运行该 warp 所需的寄存器、硬件 warp 以及共享存储都会被释放，所以需要实时更新 SM 状态寄存器表中的第二 SM 的剩余寄存器数量、剩余硬件 warp 数量以及剩余共享存储空间，以便于全局逻辑控制器及时为该 SM 下发 block。

502、当第二 SM 逻辑控制器确定运行完成的 warp 所属 block 中不存在未运行的 warp 时，确定第二 SM 中未运行完成的 block 的最高优先级，通知全局逻辑控制器更新 SM 状态寄存器表中的第二 SM 中 block 的最高优先级。

值得说明的是，当运行完成的 warp 所属 block 中还存在未运行完成的 warp 时，则第二 SM 等待下一个 warp 运行完成后，再执行步骤 501。

本发明实施例提供的 GPU 资源的分配方法，当第二逻辑控制器确定有运行完成的 warp 时，通知全局逻辑控制器更新 SM 状态寄存器表，且第二逻辑控制

器及时通知全局逻辑控制器更新 SM 状态寄存器表中的第二 SM 中 block 的最高优先级，使得全局逻辑控制器可以根据更新后的 SM 状态寄存器表，及时向 SM 下发高优先级 kernel 中的 block，提高了 SM 的资源利用率，同时加快了高优先级的 kernel 程序的响应速度。

结合图 2 至图 5 所示的 GPU 资源的分配方法，本发明实施例还提供一种 GPU 资源的分配系统，如图 6 所示，该系统包括全局逻辑控制器 601 以及至少两个能够与所述全局逻辑控制器 601 通信的流式多处理器 SM; 全局逻辑控制器包括: 第一确定单元 6011、第一查找单元 6012 以及第一分发单元 6013。

需要说明的是，该系统中的 SM 可以为能够运行至少一个完整 block 的 SM602，第一 SM603 或者第二 SM604。

其中，第一 SM603 为为能够运行至少一个 warp 的 SM，第二 SM 中 block 的最高优先级低于待分发 kernel 程序的优先级。待分发 kernel 程序为 kernel 状态寄存器表中优先级最高且未分发的 block 数量不为零的 kernel 程序。

图 6 中示出了能够运行至少一个完整 block 的 SM602，第一 SM603 以及第二 SM604，由于这三种 SM 是根据 SM 中的剩余资源量确定的，所以存在这三种 SM 可能不同时存在的情况,此外该系统中 SM 中的数量也不限于图 6 中示出的三个。

第一确定单元 6011，用于从核 kernel 状态寄存器表中确定待分发 kernel 程序，kernel 状态寄存器表中包括每个未完成运行的 kernel 程序的优先级以及每个未完成运行的 kernel 程序中未分发的线程块 block 数量。

第一查找单元 6012，用于从 SM 状态寄存器表中查找能够运行至少一个完整 block 的 SM602，SM 状态寄存器表中包括每个 SM 中的剩余资源量以及每个 SM 中 block 的最高优先级；当未查找到能够运行至少一个完整 block 的 SM602 时，从 SM 状态寄存器表中查找第一 SM603。

第一分发单元 6013，用于当第一查找单元 6012 查找到第一 SM603 时，将待分发 kernel 程序中的 block 分发给第一 SM603。

第一 SM603，用于运行第一分发单元 6013 分发的待分发 kernel 程序中的 block。

第一查找单元 6012，还用于当未查找到第一 SM603 时，查找第二 SM604。

第一分发单元 6013，还用于当第一查找单元 6012 查找到第二 SM604 时，

将待分发 kernel 程序中的 block 分发给第二 SM604。

第二 SM604，用于运行第一分发单元 6013 分发的待分发 kernel 程序中的 block。

在本发明另一实施例中，第一确定单元 6011，还用于当第一查找单元 6012 查找到能够运行至少一个完整 block 的 SM 时，确定第一数量，第一数量为能够运行一个完整 block 的 SM 实际能够运行的 block 的数量。

第一分发单元 6013，还用于当待分发 kernel 程序中未分发的 block 的数量大于第一数量时，将待分发 kernel 程序中第一数量的 block 分发给能够运行至少一个完整 block 的 SM602；当待分发 kernel 程序中未分发的 block 的数量小于或等于第一数量时，将待分发 kernel 程序中未分发的 block 全部分发给能够运行至少一个完整 block 的 SM602。

能够运行至少一个完整 block 的 SM602，用于运行第一分发单元 6013 分发的待分发 kernel 程序中的 block。

在本发明另一实施例中，如图 7 所示，第二 SM604 包括第二确定单元 6041、第二查找单元 6042、第二分发单元 6043 以及通知单元 6044。

需要说明的是，第二确定单元 6041、第二查找单元 6042、第二分发单元 6043 以及通知单元 6044 具体位于第二 SM604 中的第二 SM 逻辑控制器中。

第二确定单元 6041，用于从 block 状态寄存器表中确定优先级最高的 block，block 状态寄存器表包括被分发到第二 SM604 中的每个 block 的优先级。

第二查找单元 6042，用于查找当前的空闲硬件 warp。

第二分发单元 6043，用于确定空闲硬件 warp 能够运行一个 warp，且未接收到优先级更高的 block 时，将优先级最高的 block 中的一个 warp 分发给空闲硬件 warp。

值得说明的是，能够运行至少一个个完整 block 的 SM602 以及第一 SM603 与第二 SM604 的组成结构相同，在本发明实施例中不再一一说明。

需要说明的是，SM 状态寄存器表中包括每个 SM 的剩余寄存器数量、剩余硬件 warp 数量以及剩余共享存储空间，第一 SM603 为剩余寄存器数量大于运行一个 warp 所需的寄存器数量、剩余硬件 warp 数量大于运行一个 warp 所需的硬件 warp 数量且剩余共享存储空间大于运行一个 warp 所需的共享存储空间的 SM。

通知单元 6044，用于当确定第二 SM604 中有运行完成的 warp 时，通知全局逻辑控制器更新第二 SM604 的剩余寄存器数量、剩余 warp 数量以及剩余共享存储空间；当所确定运行完成的 warp 所属 block 中不存在未运行的 warp 时，确定第二 SM604 中未运行完成的 block 的最高优先级，通知全局逻辑控制器 601 更新 SM 状态寄存器表中的第二 SM604 中 block 的最高优先级。

本发明实施例提供的 GPU 资源的分配系统，全局逻辑控制器中的第一确定单元从 kernel 状态寄存器表中确定待分发 kernel 程序，第一查找单元从 SM 状态寄存器表中查找能够运行至少一个完整 block 的 SM，当未查找到能够运行至少一个 block 的 SM 时，则继续查找能够运行至少一个 warp 的第一 SM，第一分发单元将待分发 kernel 程序中的一个 block 分发给第一 SM，当未查找到第一 SM 时，将待分发 kernel 程序中的 block 分发给第二 SM。与现有技术中必须等待 GPU 中有空闲的 SM 时，才能将高优先级 kernel 中的 block 分发给该 SM 而导致高优先级的 kernel 程序得不到及时响应相比，本发明实施例中，当未查找到能够运行至少一个 block 的 SM 时，不是等待其他 kernel 程序释放资源，而是查找能够运行至少一个 warp 的第一 SM，由于 warp 比 block 小，所以运行完一个 warp 比运行完一个 block 更快，所以更容易查找到能够运行至少一个 warp 的 SM，查找到之后就可以将待分发 kernel 程序的一个 block 分发给第一 SM，无需等待低优先级的 kernel 程序运行完一个 block，当未查找到能够运行至少一个 warp 的第一 SM 时，将待分发 kernel 程序中的 block 分发给第二 SM，减少了分发 block 的等待时间，提高了高优先级的 kernel 程序的响应速度。

本发明实施例还提供一种 GPU 资源的分配装置，如图 8 所示，该装置中包括全局逻辑控制器以及至少两个能够与全局逻辑控制器通信的 SM。其中，SM 可以为能够运行至少一个完整 block 的 SM 或者第一 SM。全局逻辑控制器可包括存储器 81、收发器 82、处理器 83 和总线 84，其中，存储器 81、收发器 82、处理器 83 通过总线 84 通信连接。

存储器 81 可以是只读存储器（Read Only Memory，ROM），静态存储设备，动态存储设备或者随机存取存储器（Random Access Memory，RAM）。存储器 81 可以存储操作系统和其他应用程序。在通过软件或者固件来实现本发明实施例提供的技术方案时，用于实现本发明实施例提供的技术方案的程序代码保存在存储器 81 中，并由处理器 83 来执行。

收发器 82 用于装置与其他设备或通信网络（例如但不限于以太网，无线接入网(Radio Access Network,RAN),无线局域网(Wireless Local Area Network,WLAN）等）之间的通信。

处理器 83 可以采用通用的中央处理器（Central Processing Unit，CPU),微处理器，应用专用集成电路（Application Specific Integrated Circuit,ASIC),或者一个或多个集成电路，用于执行相关程序，以实现本发明实施例所提供的技术方案。

总线 84 可包括一通路，在装置各个部件（例如存储器 81、收发器 82 和处理器 83）之间传送信息。

应注意，尽管图 8 所示的硬件仅仅示出了存储器 81、收发器 82 和处理器 83 以及总线 84，但是在具体实现过程中，本领域的技术人员应当明白，该终端还包含实现正常运行所必须的其他器件。同时，根据具体需要，本领域的技术人员应当明白，还可包含实现其他功能的硬件器件。

具体的，图 8 所示的全局逻辑控制器用于实现图 6 实施例所示的系统时，该装置中的处理器 83，与存储器 81 和收发器 82 耦合，用于控制程序指令的执行，具体用于从核 kernel 状态寄存器表中确定待分发 kernel 程序，kernel 状态寄存器表中包括每个未完成运行的 kernel 程序的优先级以及每个未完成运行的 kernel 程序中未分发的线程块 block 数量，待分发 kernel 程序为 kernel 状态寄存器表中优先级最高且未分发的 block 数量不为零的 kernel 程序;从 SM 状态寄存器表中查找能够运行至少一个完整 block 的 SM，SM 状态寄存器表中包括每个 SM 中的剩余资源量以及每个 SM 中 block 的最高优先级；当未查找到能够运行至少一个完整 block 的 SM 时，从 SM 状态寄存器表中查找第一 SM，第一 SM 为能够运行至少一个线程束 warp 的 SM。

收发器 82，用于当查找到第一 SM 时，将待分发 kernel 程序中的 block 分发给第一 SM。

处理器 83，还用于当未查找到第一 SM 时，查找第二 SM，第二 SM 中 block 的最高优先级低于待分发 kernel 程序的优先级。

收发器 82，还用于当查找到第二 SM 时，将待分发 kernel 程序中的 block 分发给第二 SM。

存储器 81，还用于存储 kernel 状态寄存器表和 SM 状态寄存器表。

处理器 83，还用于当查找到能够运行至少一个完整 block 的 SM 时，确定第一数量，第一数量为能够运行一个完整 block 的 SM 实际能够运行的 block 的数量。

收发器 82，还用于当待分发 kernel 程序中未分发的 block 的数量大于第一数量时，将待分发 kernel 程序中第一数量的 block 分发给能够运行至少一个完整 block 的 SM；当待分发 kernel 程序中未分发的 block 的数量小于或等于第一数量时，将待分发 kernel 程序中未分发的 block 全部分发给能够运行至少一个完整 block 的 SM。

在本发明在本发明另一实施例中，如图 9 所示，第二 SM 包括存储器 91、收发器 92、处理器 93 和总线 94，其中，存储器 91、收发器 92、处理器 93 通过总线 94 通信连接。

存储器 91 可以是只读存储器（Read Only Memory，ROM），静态存储设备，动态存储设备或者随机存取存储器（Random Access Memory，RAM）。存储器 91 可以存储操作系统和其他应用程序。在通过软件或者固件来实现本发明实施例提供的技术方案时，用于实现本发明实施例提供的技术方案的程序代码保存在存储器 91 中，并由处理器 93 来执行。

收发器 92 用于装置与其他设备或通信网络（例如但不限于以太网，无线接入网（Radio Access Network，RAN），无线局域网（Wireless Local Area Network，WLAN）等）之间的通信。

处理器 93 可以采用通用的中央处理器（Central Processing Unit，CPU），微处理器，应用专用集成电路（Application Specific Integrated Circuit，ASIC），或者一个或多个集成电路，用于执行相关程序，以实现本发明实施例所提供的技术方案。

总线 94 可包括一通路，在装置各个部件（例如存储器 91、收发器 92 和处理器 93）之间传送信息。

应注意，尽管图 9 所示的硬件仅仅示出了存储器 91、收发器 92 和处理器 93 以及总线 94，但是在具体实现过程中，本领域的技术人员应当明白，该终端还包含实现正常运行所必须的其他器件。同时，根据具体需要，本领域的技术人员应当明白，还可包含实现其他功能的硬件器件。

具体的，图 9 所示的第一 SM 用于实现图 6 和图 7 实施例所示的系统时，该

装置中的处理器 93，与存储器 91 和收发器 92 耦合，用于控制程序指令的执行，具体用于从 block 状态寄存器表中确定优先级最高的 block，block 状态寄存器表包括被分发到第二 SM 中的每个 block 的优先级；查找当前的空闲硬件 warp。

收发器 92，还用于当确定空闲硬件 warp 能够运行一个 warp，且未接收到优先级更高的 block 时，将优先级最高的 block 中的一个 warp 分发给空闲硬件 warp。

值得说明的是，SM 状态寄存器表中包括每个 SM 的剩余寄存器数量、剩余硬件 warp 数量以及剩余共享存储空间，第一 SM 为剩余寄存器数量大于运行一个 warp 所需的寄存器数量、剩余硬件 warp 数量大于运行一个 warp 所需的硬件 warp 数量且剩余共享存储空间大于运行一个 warp 所需的共享存储空间的 SM。

收发器 92，还用于当确定第二 SM 中有运行完成的 warp 时，通知全局逻辑控制器更新第二 SM 的剩余寄存器数量、剩余 warp 数量以及剩余共享存储空间；当确定运行完成的 warp 所属 block 中不存在未运行的 warp 时，确定第二 SM 中未运行完成的 block 的最高优先级，通知全局逻辑控制器更新 SM 状态寄存器表中的第二 SM 中 block 的最高优先级。

本发明实施例提供的 GPU 资源的分配装置，全局逻辑控制器中的处理器从 kernel 状态寄存器表中确定待分发 kernel 程序，从 SM 状态寄存器表中查找能够运行至少一个完整 block 的 SM，当未查找到能够运行至少一个 block 的 SM 时，则继续查找能够运行至少一个 warp 的第一 SM，收发器将待分发 kernel 程序中的一个 block 分发给第一 SM，当未查找到第一 SM 时，将待分发 kernel 程序中的 block 分发给第二 SM。与现有技术中必须等待 GPU 中有空闲的 SM 时，才能将高优先级 kernel 中的 block 分发给该 SM 而导致高优先级的 kernel 程序得不到及时响应相比，本发明实施例中，当未查找到能够运行至少一个 block 的 SM 时，不是等待其他 kernel 程序释放资源，而是查找能够运行至少一个 warp 的第一 SM，由于 warp 比 block 小，所以运行完一个 warp 比运行完一个 block 更快，所以更容易查找到能够运行至少一个 warp 的 SM，查找到之后就可以将待分发 kernel 程序的一个 block 分发给第一 SM，无需等待低优先级的 kernel 程序运行完一个 block，当未查找到能够运行至少一个 warp 的第一 SM 时，将待分发 kernel 程序中的 block 分发给第二 SM，减少了分发 block 的等待时间，提高了高优先级的 kernel 程序的响应速度。

所属领域的技术人员可以清楚地了解到，为描述的方便和简洁，仅以上述各功能模块的划分进行举例说明，实际应用中，可以根据需要而将上述功能分配由不同的功能模块完成，即将装置的内部结构划分成不同的功能模块，以完成以上描述的全部或者部分功能。上述描述的系统，装置和单元的具体工作过程，可以参考前述方法实施例中的对应过程，在此不再赘述。

在本申请所提供的几个实施例中，应该理解到，所揭露的系统，装置和方法，可以通过其它的方式实现。例如，以上所描述的装置实施例仅仅是示意性的，例如，所述模块或单元的划分，仅仅为一种逻辑功能划分，实际实现时可以有另外的划分方式，例如多个单元或组件可以结合或者可以集成到另一个系统，或一些特征可以忽略，或不执行。另一点，所显示或讨论的相互之间的耦合或直接耦合或通信连接可以是通过一些接口，装置或单元的间接耦合或通信连接，可以是电性，机械或其它的形式。

所述作为分离部件说明的单元可以是或者也可以不是物理上分开的，作为单元显示的部件可以是或者也可以不是物理单元，即可以位于一个地方，或者也可以分布到多个网络单元上。可以根据实际的需要选择其中的部分或者全部单元来实现本实施例方案的目的。

另外，在本发明各个实施例中的各功能单元可以集成在一个处理单元中，也可以是各个单元单独物理存在，也可以两个或两个以上单元集成在一个单元中。上述集成的单元既可以采用硬件的形式实现，也可以采用软件功能单元的形式实现。

所述集成的单元如果以软件功能单元的形式实现并作为独立的产品销售或使用时，可以存储在一个计算机可读取存储介质中。基于这样的理解，本发明的技术方案本质上或者说对现有技术做出贡献的部分或者该技术方案的全部或部分可以以软件产品的形式体现出来，该计算机软件产品存储在一个存储介质中，包括若干指令用以使得一台计算机设备（可以是个人计算机，服务器，或者网络设备等）或处理器（processor）执行本发明各个实施例所述方法的全部或部分步骤。而前述的存储介质包括：U 盘、移动硬盘、只读存储器（ROM，Read-Only Memory）、随机存取存储器（RAM，Random Access Memory）、磁碟或者光盘等各种可以存储程序代码的介质。

以上所述，仅为本发明的具体实施方式，但本发明的保护范围并不局限于

此，任何熟悉本技术领域的技术人员在本发明揭露的技术范围内，可轻易想到变化或替换，都应涵盖在本发明的保护范围之内。因此，本发明的保护范围应以所述权利要求的保护范围为准。

权 利 要 求 书

1、一种图形处理器 GPU 资源的分配方法，其特征在于，所述方法应用于 GPU 资源的分配系统中，所述系统包括全局逻辑控制器以及至少两个能够与所述全局逻辑控制器通信的流式多处理器 SM，所述方法包括:

所述全局逻辑控制器从核 kernel 状态寄存器表中确定待分发 kernel 程序，所述 kernel 状态寄存器表中包括每个未完成运行的 kernel 程序的优先级以及每个未完成运行的 kernel 程序中未分发的线程块 block 数量，所述待分发 kernel 程序为所述 kernel 状态寄存器表中优先级最高且未分发的 block 数量不为零的 kernel 程序;

所述全局逻辑控制器从 SM 状态寄存器表中查找能够运行至少一个完整 block 的 SM，所述 SM 状态寄存器表中包括每个 SM 中的剩余资源量以及每个 SM 中 block 的最高优先级;

当所述全局逻辑控制器未查找到能够运行至少一个完整 block 的 SM 时，从所述 SM 状态寄存器表中查找第一 SM，所述第一 SM 为能够运行至少一个线程束 warp 的 SM;

当所述全局逻辑控制器查找到所述第一 SM 时，将所述待分发 kernel 程序中的 block 分发给所述第一 SM;

当所述全局逻辑控制器未查找到所述第一 SM 时，查找第二 SM，所述第二 SM 中 block 的最高优先级低于所述待分发 kernel 程序的优先级;

当所述全局逻辑控制器查找到所述第二 SM 时，将所述待分发 kernel 程序中的 block 分发给所述第二 SM。

2、根据权利要求 1 所述的 GPU 资源的分配方法，其特征在于，在所述全局逻辑控制器从 SM 状态寄存器表中查找能够运行至少一个完整 block 的 SM 之后，所述方法还包括:

当所述全局逻辑控制器查找到能够运行至少一个完整 block 的 SM 时，确定第一数量，所述第一数量为所述能够运行一个完整 block 的 SM 实际能够运行的 block 的数量;

当所述待分发 kernel 程序中未分发的 block 的数量大于所述第一数量时，将所述待分发 kernel 程序中所述第一数量的 block 分发给所述能够运行至少一个完整 block 的 SM;

当所述待分发kernel程序中未分发的block的数量小于或等于所述第一数量时，将所述待分发kernel程序中未分发的block全部分发给所述能够运行至少一个完整block的SM。

3、根据权利要求2所述的GPU资源的分配方法，其特征在于，在所述全局逻辑控制器将所述待分发kernel程序中的block分发给所述第二SM之后，所述方法还包括:

第二SM逻辑控制器从block状态寄存器表中确定优先级最高的block，所述第二SM逻辑控制器为所述第二SM中的SM逻辑控制器，所述block状态寄存器表包括被分发到所述第二SM中的每个block的优先级;

所述第二SM逻辑控制器查找当前的空闲硬件warp;

当所述第二SM逻辑控制器确定所述空闲硬件warp能够运行一个warp，且未接收到优先级更高的block时，将所述优先级最高的block中的一个warp分发给所述空闲硬件warp。

4、根据权利要求1至3中任一项所述的GPU资源的分配方法，其特征在于，所述SM状态寄存器表中包括每个SM的剩余寄存器数量、剩余硬件warp数量以及剩余共享存储空间，所述第一SM为所述剩余寄存器数量大于运行一个warp所需的寄存器数量、所述剩余硬件warp数量大于运行一个warp所需的硬件warp数量且所述剩余共享存储空间大于运行一个warp所需的共享存储空间的SM。

5、根据权利要求4所述的GPU资源的分配方法，其特征在于，在所述当所述第二SM逻辑控制器确定所述空闲硬件warp能够运行一个warp，且未接收到优先级更高的block时，将所述优先级最高的block中的一个warp分发给所述硬件warp之后，所述方法还包括:

所述第二SM逻辑控制器确定所述第二SM中有运行完成的warp时，通知所述全局逻辑控制器更新所述第二SM的剩余寄存器数量、剩余warp数量以及剩余共享存储空间;

当所述第二SM逻辑控制器确定所述运行完成的warp所属block中不存在未运行的warp时，确定所述第二SM中未运行完成的block的最高优先级，通知所述全局逻辑控制器更新所述SM状态寄存器表中的所述第二SM中block的最高优先级。

6、一种图形处理器 GPU 资源的分配系统，其特征在于，所述系统包括全局逻辑控制器以及至少两个能够与所述全局逻辑控制器通信的流式多处理器 SM；所述全局逻辑控制器包括：第一确定单元、第一查找单元以及第一分发单元；

所述第一确定单元，用于从核 kernel 状态寄存器表中确定待分发 kernel 程序，所述 kernel 状态寄存器表中包括每个未完成运行的 kernel 程序的优先级以及每个未完成运行的kernel程序中未分发的线程块block数量，所述待分发kernel程序为所述 kernel 状态寄存器表中优先级最高且未分发的 block 数量不为零的 kernel 程序；

所述第一查找单元，用于从 SM 状态寄存器表中查找能够运行至少一个完整 block 的 SM，所述 SM 状态寄存器表中包括每个 SM 中的剩余资源量以及每个 SM 中 block 的最高优先级；当未查找到能够运行至少一个完整 block 的 SM 时，从所述 SM 状态寄存器表中查找第一 SM，所述第一 SM 为能够运行至少一个线程束 warp 的 SM；

所述第一分发单元，用于当所述第一查找单元查找到所述第一 SM 时，将所述待分发 kernel 程序中的 block 分发给所述第一 SM；

所述第一 SM，用于运行所述第一分发单元分发的所述待分发 kernel 程序中的 block；

所述第一查找单元，还用于当未查找到所述第一 SM 时，查找第二 SM，所述第二 SM 中 block 的最高优先级低于所述待分发 kernel 程序的优先级；

所述第一分发单元，还用于当所述第一查找单元查找到所述第二 SM 时，将所述待分发 kernel 程序中的 block 分发给所述第二 SM；

所述第二 SM，用于运行所述第一分发单元分发的所述待分发 kernel 程序中的 block。

7、根据权利要求 6 所述的 GPU 资源的分配系统，其特征在于，

所述第一确定单元，还用于当所述第一查找单元查找到能够运行至少一个完整 block 的 SM 时，确定第一数量，所述第一数量为所述能够运行一个完整 block 的 SM 实际能够运行的 block 的数量；

所述第一分发单元，还用于当所述待分发 kernel 程序中未分发的 block 的数量大于所述第一数量时，将所述待分发 kernel 程序中所述第一数量的 block 分发

给所述能够运行至少一个完整 block 的 SM；当所述待分发 kernel 程序中未分发的 block 的数量小于或等于所述第一数量时，将所述待分发 kernel 程序中未分发的 block 全部分发给所述能够运行至少一个完整 block 的 SM；

所述能够运行至少一个完整 block 的 SM，用于运行所述第一分发单元分发的所述待分发 kernel 程序中的 block。

8、根据权利要求 7 所述的 GPU 资源的分配系统，其特征在于，所述第二 SM 包括第二确定单元、第二查找单元以及第二分发单元；

所述第二确定单元，用于从 block 状态寄存器表中确定优先级最高的 block，所述 block 状态寄存器表包括被分发到所述第二 SM 中的每个 block 的优先级；

所述第二查找单元，用于查找当前的空闲硬件 warp；

所述第二分发单元，用于确定所述空闲硬件 warp 能够运行一个 warp，且未接收到优先级更高的 block 时，将所述优先级最高的 block 中的一个 warp 分发给所述空闲硬件 warp。

9、根据权利要求 6 至 8 中任一项所述的 GPU 资源的分配系统，其特征在于，所述 SM 状态寄存器表中包括每个 SM 的剩余寄存器数量、剩余硬件 warp 数量以及剩余共享存储空间，所述第一 SM 为所述剩余寄存器数量大于运行一个 warp 所需的寄存器数量、所述剩余硬件 warp 数量大于运行一个 warp 所需的硬件 warp 数量且所述剩余共享存储空间大于运行一个 warp 所需的共享存储空间的 SM。

10、根据权利要求 9 所述的 GPU 资源的分配系统，其特征在于，所述第二 SM 中还包括通知单元；

所述通知单元，用于当确定所述第二 SM 中有运行完成的 warp 时，通知所述全局逻辑控制器更新所述第二 SM 的剩余寄存器数量、剩余 warp 数量以及剩余共享存储空间；当所确定所述运行完成的 warp 所属 block 中不存在未运行的 warp 时，确定所述第二 SM 中未运行完成的 block 的最高优先级，通知所述全局逻辑控制器更新所述 SM 状态寄存器表中的所述第二 SM 中 block 的最高优先级。

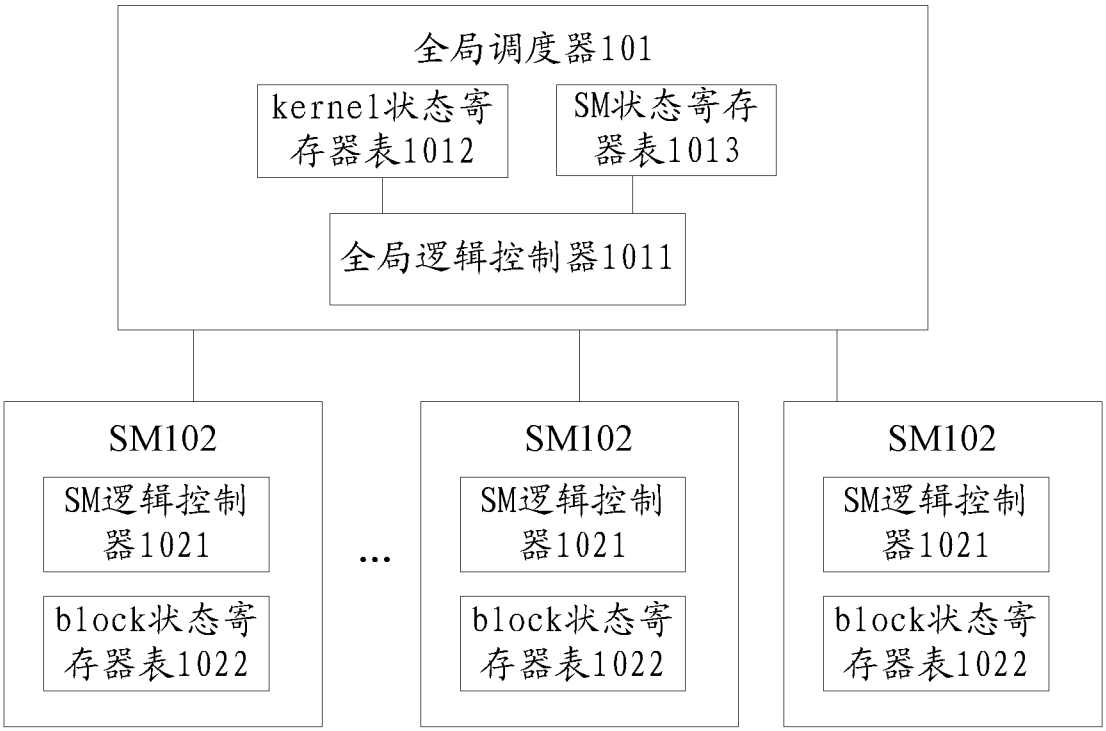
全局调度器101
kernel状态寄存器表1012
SM状态寄存器表1013
全局逻辑控制器1011
SM102
SM逻辑控制器1021
block状态寄存器表1022
...
SM102
SM逻辑控制器1021
block状态寄存器表1022
SM102
SM逻辑控制器1021
block状态寄存器表1022

图 1


全局逻辑控制器从kernel状态寄存器表中确定待分发kernel程序
201
全局逻辑控制器从SM状态寄存器表中查找能够运行至少一个完整block的SM，SM状态寄存器表中包括每个SM中的剩余资源量以及每个SM中block的最高优先级
202
当全局逻辑控制器未查找到能够运行至少一个完整block的SM时，从SM状态寄存器表中查找第一SM，第一SM为能够运行至少一个warp的SM
203
当全局逻辑控制器查找到第一SM时，将待分发kernel程序中的block分发给第一SM
204
当全局逻辑控制器未查找到第一SM时，查找第二SM，第二SM中block的最高优先级低于待分发kernel程序的优先级
205
当全局逻辑控制器查找到第二SM时，将待分发kernel程序中的block分发给第二SM
206

图 2


全局逻辑控制器从kernel状态寄存器表中确定待分发kernel程序
201
全局逻辑控制器从SM状态寄存器表中查找能够运行至少一个完整block的SM，SM状态寄存器表中包括每个SM中的剩余资源量以及每个SM中block的最高优先级
202
当全局逻辑控制器未查找到能够运行至少一个完整block的SM时，从SM状态寄存器表中查找第一SM，第一SM为能够运行至少一个warp的SM
203
当全局逻辑控制器查找到第一SM时，将待分发kernel程序中的block分发给第一SM
204
当全局逻辑控制器未查找到第一SM时，查找第二SM，第二SM中block的最高优先级低于待分发kernel程序的优先级
205
当全局逻辑控制器查找到第二SM时，将待分发kernel程序中的block分发给第二SM
206
当全局逻辑控制器查找到能够运行至少一个完整block的SM时，确定第一数量，第一数量为能够运行一个完整block的SM实际能够运行的block的数量
207
当待分发kernel程序中未分发的block的数量大于第一数量时，将待分发kernel程序中第一数量的block分发给能够运行至少一个完整block的SM
208
当待分发kernel程序中未分发的block的数量小于或等于第一数量时，将待分发kernel程序中的未分发的block全部分发给能够运行至少一个完整block的SM
209

图 3

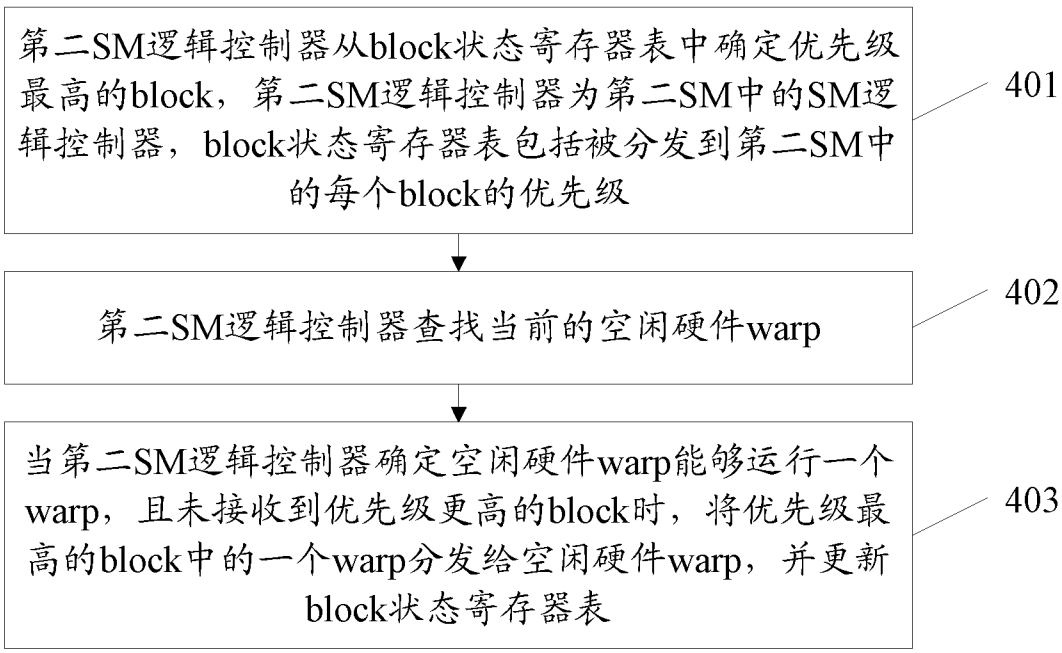
第二SM逻辑控制器从block状态寄存器表中确定优先级最高的block，第二SM逻辑控制器为第二SM中的SM逻辑控制器，block状态寄存器表包括被分发到第二SM中的每个block的优先级
401
第二SM逻辑控制器查找当前的空闲硬件warp
402
当第二SM逻辑控制器确定空闲硬件warp能够运行一个warp，且未接收到优先级更高的block时，将优先级最高的block中的一个warp分发给空闲硬件warp，并更新block状态寄存器表
403

图 4

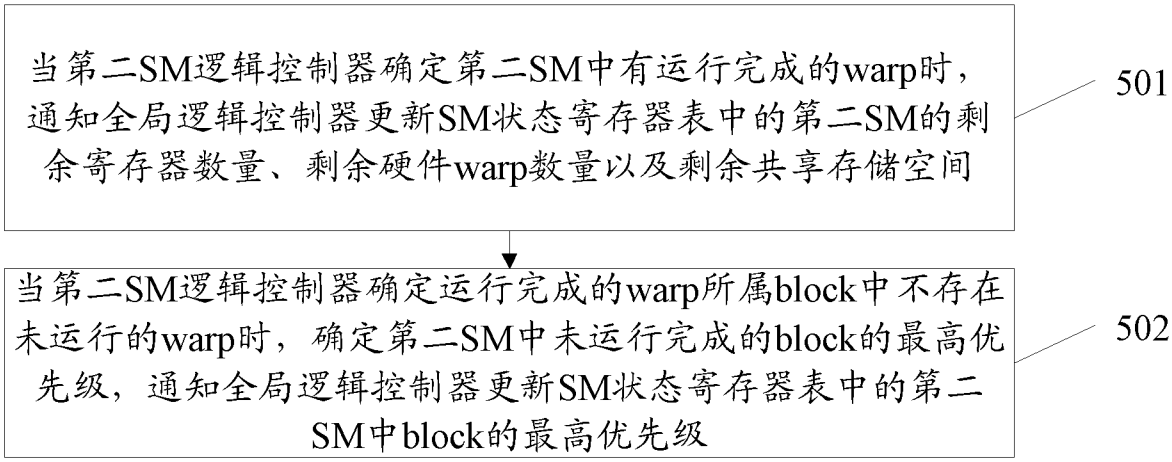
当第二SM逻辑控制器确定第二SM中有运行完成的warp时，通知全局逻辑控制器更新SM状态寄存器表中的第二SM的剩余寄存器数量、剩余硬件warp数量以及剩余共享存储空间
501
当第二SM逻辑控制器确定运行完成的warp所属block中不存在未运行的warp时，确定第二SM中未运行完成的block的最高优先级，通知全局逻辑控制器更新SM状态寄存器表中的第二SM中block的最高优先级
502

图 5

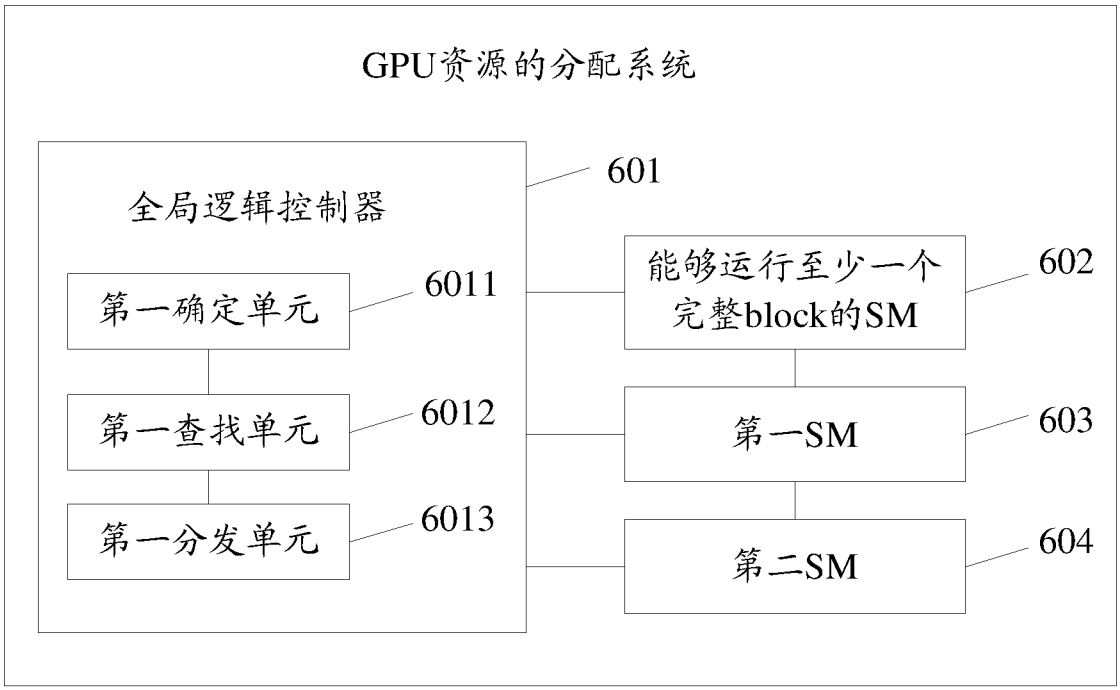
GPU资源的分配系统
全局逻辑控制器
601
第一确定单元
6011
第一查找单元
6012
第一分发单元
6013
能够运行至少一个完整block的SM
602
第一SM
603
第二SM
604

图 6

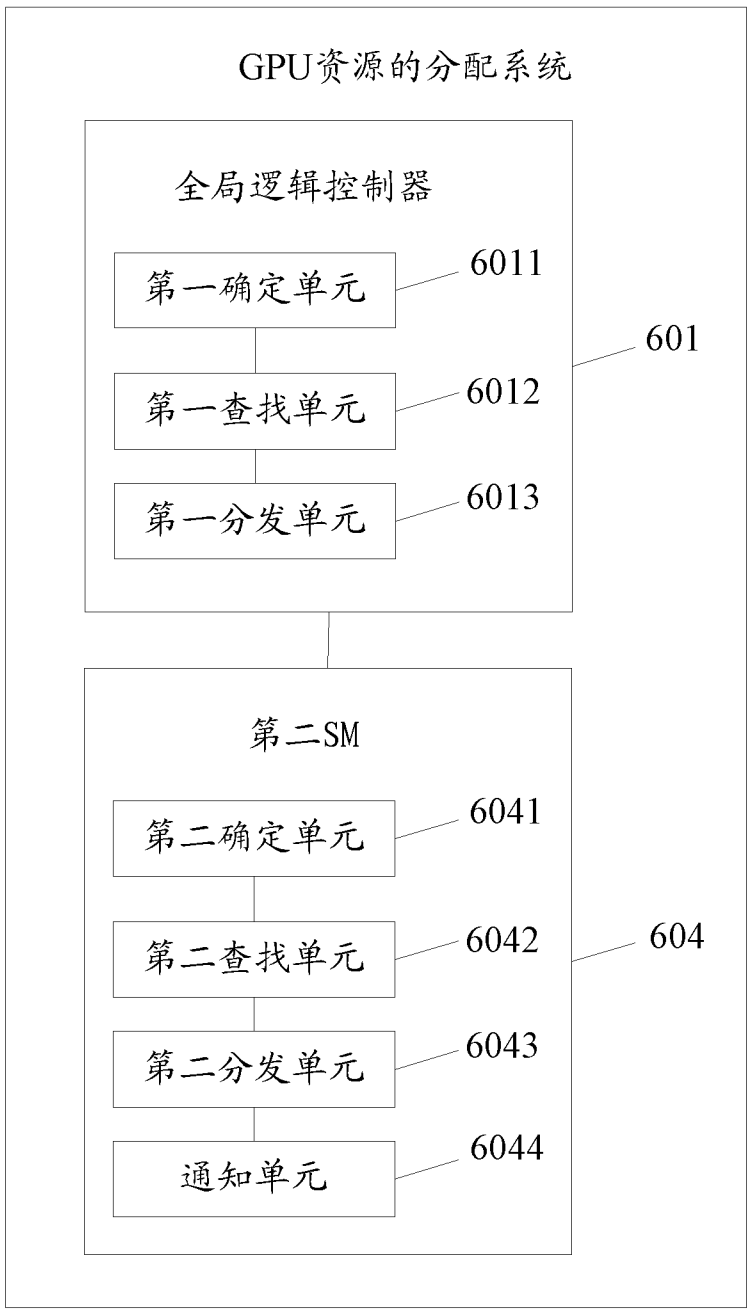
GPU资源的分配系统
全局逻辑控制器
第一确定单元
6011
第一查找单元
6012
第一分发单元
6013
601
第二SM
第二确定单元
6041
第二查找单元
6042
第二分发单元
6043
通知单元
6044
604

图 7

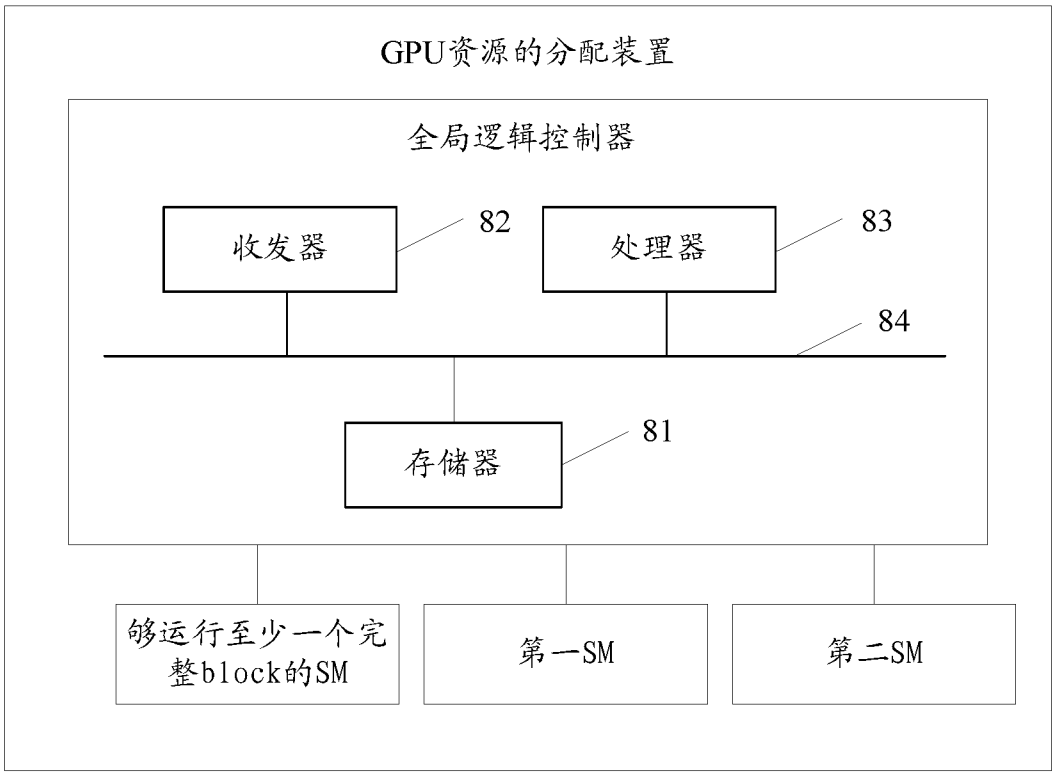
GPU资源的分配装置
全局逻辑控制器
收发器
82
处理器
83
84
存储器
81
够运行至少一个完整block的SM
第一SM
第二SM

图 8

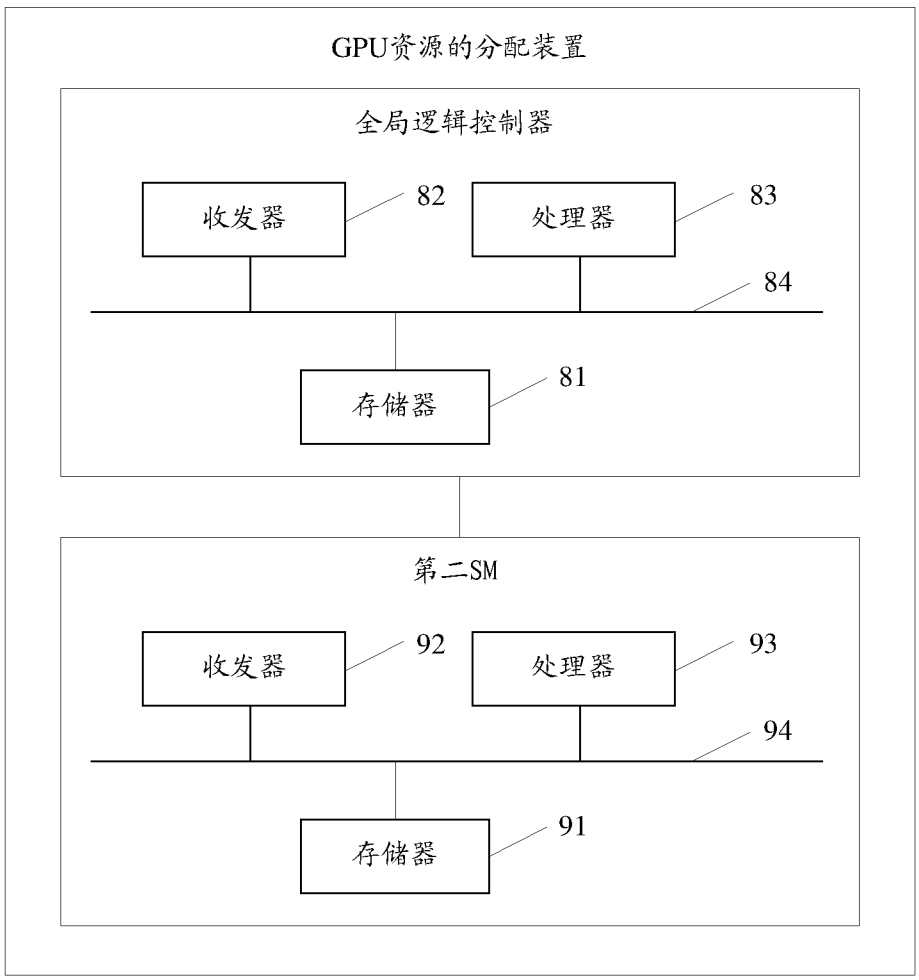
GPU资源的分配装置
全局逻辑控制器
收发器
82
处理器
83
84
存储器
81
第二SM
收发器
92
处理器
93
94
存储器
91

图 9

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2016/083315

A. CLASSIFICATION OF SUBJECT MATTER

G06F 9/50 (2006.01) i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

CNKI, CNPAT, WPI, EPODOC, IEEE: GPU, graphic processing unit, resource, block, thread

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	CN 102541640 A (XIAMEN MEIYA PICO INFORMATION CO., LTD.) 04 July 2012 (04.07.2012) description, paragraphs [0041] to [0077], and figures 1 to 7	1-10
A	CN 103631660 A (DATA ASSURANCE AND COMM SECURITY RES CT OF CHINESE ACADEMY OF SCIENCES) 12 March 2014 (12.03.2014) the whole document	1-10
A	CN 103617088 A (ICUBE CORP.) 05 March 2014 (05.03.2014) the whole document	1-10
A	US 2008109810 A1 (MICROSOFT CORPORATION) 08 May 2008 (08.05.2008) the whole document	1-10

☐ Further documents are listed in the continuation of Box C. ☒ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&"document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
11 August 2016	29 August 2016
Name and mailing address of the ISA State Intellectual Property Office of the P. R. China No. 6, Xitucheng Road, Jimenqiao Haidian District, Beijing 100088, China Facsimile No. (86-10) 62019451	Authorized officer WANG, Li Telephone No. (86-10) 82245262

Form PCT/ISA /210 (second sheet) (July 2009)

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2016/083315

Patent Documents referred in the Report	Publication Date	Patent Family	Publication Date
CN 102541640 A	04 July 2012	CN 102541640 B	29 October 2014
CN 103631660 A	12 March 2014	None	
CN 103617088 A	05 March 2014	None	
US 2008109810 A1	08 May 2008	US 7830387 B2	09 November 2010

国际检索报告

国际申请号
PCT/CN2016/083315

A. 主题的分类

G06F 9/50(2006.01)i

按照国际专利分类(IPC)或者同时按照国家分类和IPC两种分类

B. 检索领域

检索的最低限度文献(标明分类系统和分类号)

G06F

包含在检索领域中的除最低限度文献以外的检索文献

在国际检索时查阅的电子数据库(数据库的名称，和使用的检索词（如使用））

CNKI, CNPAT, WPI, EPODOC, IEEE：图形，处理器，资源，块，线程，GPU，graphic processing unit,resource, block, thread

C. 相关文件

类 型*	引用文件，必要时，指明相关段落	相关的权利要求
A	CN 102541640 A (厦门市美亚柏科信息股份有限公司) 2012年 7月 4日 (2012-07-04) 说明书第[0041]-[0077]段，图1-7	1-10
A	CN 103631660 A (中国科学院数据与通信保护研究教育中心) 2014年 3月 12日 (2014-03-12) 全文	1-10
A	CN 103617088 A (深圳中微电科技有限公司) 2014年 3月 5日 (2014-03-05) 全文	1-10
A	US 2008109810 A1 (MICROSOFT CORPORATION) 2008年 5月 8日 (2008-05-08) 全文	1-10

☐ 其余文件在C栏的续页中列出。

☑ 见同族专利附件。

* 引用文件的具体类型：

“A” 认为不特别相关的表示了现有技术一般状态的文件

“E” 在国际申请日的当天或之后公布的在先申请或专利

“L” 可能对优先权要求构成怀疑的文件，或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件（如具体说明的）

“O” 涉及口头公开、使用、展览或其他方式公开的文件

“P” 公布日先于国际申请日但迟于所要求的优先权日的文件

“T” 在申请日或优先权日之后公布，与申请不相抵触，但为了理解发明之理论或原理的在后文件

“X” 特别相关的文件，单独考虑该文件，认定要求保护的发明不是新颖的或不具有创造性

“Y” 特别相关的文件，当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时，要求保护的发明不具有创造性

“&” 同族专利的文件

国际检索实际完成的日期	国际检索报告邮寄日期
2016年 8月 11日	2016年 8月 29日

ISA/CN的名称和邮寄地址	受权官员
中华人民共和国国家知识产权局(ISA/CN) 中国北京市海淀区蓟门桥西土城路6号 100088	王丽
传真号 (86-10)62019451	电话号码 (86-10)82245262

国际检索报告
关于同族专利的信息

国际申请号
PCT/CN2016/083315

检索报告引用的专利文件			公布日 (年/月/日)	同族专利			公布日 (年/月/日)
CN	102541640	A	2012年 7月 4日	CN	102541640	B	2014年 10月 29日
CN	103631660	A	2014年 3月 12日		无		
CN	103617088	A	2014年 3月 5日		无		
US	2008109810	A1	2008年 5月 8日	US	7830387	B2	2010年 11月 9日