



(12)发明专利

(10)授权公告号 CN 104395890 B

(45)授权公告日 2018.12.07

(21)申请号 201380033791.1

(22)申请日 2013.06.07

(65)同一申请的已公布的文献号

申请公布号 CN 104395890 A

(43)申请公布日 2015.03.04

(30)优先权数据

61/657,404 2012.06.08 US

(85)PCT国际申请进入国家阶段日

2014.12.25

(86)PCT国际申请的申请数据

PCT/US2013/044682 2013.06.07

(87)PCT国际申请的公布数据

W02013/185015 EN 2013.12.12

(73)专利权人 超威半导体公司

地址 美国加利福尼亚州

(72)发明人 亚历山大·洛希夫斯基

(74)专利代理机构 上海胜康律师事务所 31263

代理人 樊英如 李献忠

(51)Int.Cl.

G06F 9/50(2006.01)

G06F 17/30(2006.01)

(56)对比文件

CN 102156658 A,2011.08.17,

US 7554959 B1,2009.06.30,

CN 1777918 A,2006.05.24,

US 2012069029 A1,2012.03.22,

Jeff A.Stuart,Michael Cox,John

D.Owens.GPU-to-CPU Callbacks.《Euro-Par

2010 Parallel Processing Workshops》.2010,

365-372.

审查员 邵娜娜

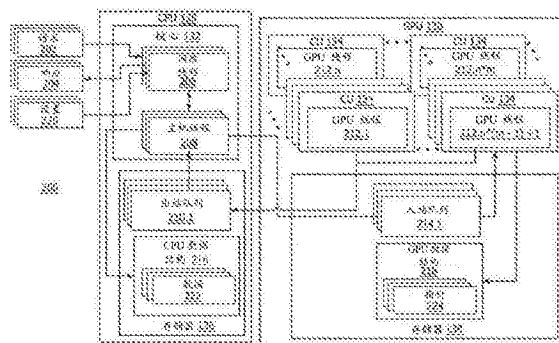
权利要求书4页 说明书7页 附图10页

(54)发明名称

使用异构处理器为应用程序提供低潜伏时间的系统和方法

(57)摘要

本发明公开用于响应请求的方法、装置和计算机可读介质。一种响应请求的方法可包括接收包括回调函数的请求。所述一个或多个请求可接收于与第一类型的处理器相关联的第一存储器中,所述第一类型的处理器可为CPU。所述请求可被移动至第二存储器。所述第二存储器可与第二类型的处理器相关联,所述第二类型的处理器可为GPU。GPU线程可在所述请求的数量为至少阈值数量时处理所述请求,以便确定所述请求的结果。所述方法可包括将所述结果移动至所述第一存储器。所述方法可包括所述CPU利用所述对应结果来执行所述一个或多个回调函数。GPU持久线程可检查请求的数量,以便确定何时达到请求的阈值数量。



1. 一种用于在多处理器计算机系统中响应请求的方法,所述方法包括:

由第一类型的一个或多个处理器接收一个或多个请求,每个请求具有密钥值,其中所述第一类型的一个或多个处理器与第一存储器相关联,并且所述第一存储器包括具有多个数据值的第一数据结构,所述每个数据值对应于所述第一存储器中的多个位置中的一个位置;

将所述一个或多个请求从与所述第一类型的所述一个或多个处理器相关联的所述第一存储器移动至与第二类型的一个或多个处理器相关联的第二存储器,其中所述第二存储器包括第二数据结构,并且所述第二数据结构包括:

(i) 多个对值,每个对值包括多个密钥值中的一个以及所述第一存储器中的所述多个位置中的一个,以及

(ii) 多个索引值,其各自对应于所述多个对值中的一个或多个,每个指针包括索引值;

存储在所述第二数据结构中,在每个索引值处,所述多个对值中的一个或多个各自具有在所述第一存储器中的所述位置中的存储对应数据值的一个位置;

基于所述一个或多个请求中的至少一个,通过所述第二类型的所述一个或多个处理器修改在所述第二存储器中的所述第二数据结构;

由所述第二类型的所述一个或多个处理器确定所述一个或多个请求中的每一个的一个或多个确定的指针,其中所述指针基于所述请求中的信息来确定所述一个或多个确定的指针中的每一个对应于所述密钥值和所述第一存储器中的数据值的对应位置,其中所述一个或多个确定的指针中的每一个包括由每个相应请求中的所述密钥值确定的指针值;

将所述密钥值和对应位置移动至所述第一存储器;

对于所述一个或多个确定的指针中的每一个,通过所述第一类型的所述一个或多个处理器检索由所述一个或多个确定的指针所指向的一个或多个数据值的检索数据,其中所述检索数据是在所述第一存储器中在所述对应位置处从所述第一存储器中的所述第一数据结构检索的;以及

通过所述第一类型的所述一个或多个处理器发送所述检索数据来响应所述一个或多个请求。

2. 如权利要求1所述的方法,其中所述第一类型的所述一个或多个处理器包括一个或多个中央处理器单元核心,并且所述第二类型的所述一个或多个处理器包括一个或多个图形处理单元核心。

3. 如权利要求2所述的方法,其中所述一个或多个图形处理单元核心中的每一个正基于图形处理单元核心运行持久线程。

4. 如权利要求2所述的方法,其中所述一个或多个图形处理单元核心访问所述第二存储器可快于所述一个或多个中央处理器单元核心访问所述第二存储器。

5. 如权利要求2所述的方法,其中所述一个或多个中央处理器单元核心对所述第一存储器的第一访问时间小于所述一个或多个图形处理单元核心对所述第一存储器的第二访问时间。

6. 如权利要求1所述的方法,其还包括:

由所述第一类型的所述一个或多个处理器接收设置请求;以及

由所述第一类型的所述一个或多个处理器基于所述设置请求修改所述第一数据结构。

7. 如权利要求1所述的方法,其中所述移动还包括:

当多个请求达到阈值数量时将所述一个或多个请求从所述第一存储器移动至所述第二存储器。

8. 如权利要求1所述的方法,其中所述多个数据值的一个数据值的大小大于所述一个或多个确定的指针的一个指针值的大小。

9. 一种在多处理器计算机系统中响应请求的方法,所述方法包括:

接收一个或多个请求,所述一个或多个请求各自包括回调函数并且具有密钥值,其中所述一个或多个请求被接收于与一个或多个中央处理器单元核心相关联的第一存储器中,其中所述第一存储器包括具有多个数据值的第一数据结构,每个数据值对应于所述第一存储器中的多个位置中的一个位置;

将所述一个或多个请求移动至第二存储器,其中所述第二存储器与一个或多个图形处理单元核心相关联并且所述第二存储器包括第二数据结构,并且所述第二数据结构包括:

(i) 多个对值,每个对值包括多个密钥值中的一个以及所述第一存储器中的所述多个位置中的一个,以及

(ii) 多个索引值,其各自对应于所述多个对值中的一个或多个,每个指针包括索引值;

存储在所述第二数据结构中,在每个索引值处,所述多个对值中的一个或多个各自具有在所述第一存储器中的所述位置中的存储相应数据值的一个位置;

基于所述一个或多个请求中的至少一个,由所述一个或多个图形处理单元核心修改所述第二存储器中的所述第二数据结构;

当请求的数量为至少阈值数量时确定所述一个或多个请求中的每一个的一个或多个确定的指针,其中所述一个或多个确定的指针中的每一个对应于所述密钥值和所述第一存储器中的数据值的对应位置,其中所述一个或多个确定的指针中的每一个包括由每个对应请求中的所述密钥值确定的指针值;

将所述密钥值和所述对应位置移动至所述第一存储器;

对于所述一个或多个确定的指针中的每一个,通过所述一个或多个中央处理器单元核心检索由所述一个或多个确定的指针所指向的一个或多个数据值的检索数据,其中所述检索数据是在所述第一存储器中的所述对应位置处从所述第一存储器检索的;以及

通过所述一个或多个中央处理器单元核心,使用所述一个或多个请求的所述回调函数,用所述检索数据来执行所述一个或多个请求中的每一个。

10. 如权利要求9所述的方法,其中与一个或多个中央处理器单元核心相比,一个或多个图形处理单元核心的对所述第二存储器的访问时间较少。

11. 如权利要求9所述的方法,其中将所述一个或多个请求移动至所述第二存储器包括:

将所述请求移动至所述第二存储器中的入站队列。

12. 如权利要求11所述的方法,其中当所述请求的所述数量为至少所述阈值数量并且所述图形处理单元包括一个或多个图形处理单元线程时,所述方法还包括:

通过所述一个或多个图形处理单元线程中的图形处理单元线程检查所述第二存储器中的所述入站队列,以确定所述请求的所述数量何时为至少所述阈值数量,其中所述图形处理单元线程为持久图形处理单元线程。

13. 如权利要求9所述的方法,其中移动所述检索数据至所述第一存储器包括:

移动所述检索数据到所述第一存储器的出站队列。

14. 一种用于响应请求的多处理器计算机系统,其包括:

第一类型的一个或多个处理器,其中所述第一类型的一个或多个处理器与第一存储器相关联,并且所述第一存储器包括具有多个数据值的第一数据结构,每个数据值对应于所述第一存储器中的多个位置中的一个位置;

一个或多个第二类型的处理器,其中所述第二类型的所述一个或多个处理器与第二存储器相关联;以及

所述第二存储器包括第二数据结构,并且所述第二数据结构包括:

(i) 多个对值,每个对值包括多个密钥值中的一个以及所述第一存储器中的多个位置中的一个,以及

(ii) 多个索引值,其各自对应于所述多个对值中的一个或多个,每个指针包括索引值;

存储在所述第二数据结构中,在每个索引值处,所述多个对值中的一个或多个各自具有在所述第一存储器中的所述位置中的存储对应数据值的一个位置;

其中所述第一类型的一个或多个处理器被配置来:

接收包括回调函数且各自具有密钥值的一个或多个请求,并且

将所述一个或多个请求移动至所述第二存储器,

所述第二类型的所述一个或多个处理器被配置来:

基于所述一个或多个请求中的至少一个来修改所述第二存储器中的所述第二数据结构;

当所述请求的数量为至少阈值数量时,确定所述一个或多个请求中的每一个的一个或多个确定的指针,其中所述一个或多个确定的指针中的每一个对应于所述密钥值和所述第一存储器中的数据值的对应位置,其中所述一个或多个确定的指针中的每一个包括由每个对应请求中的所述密钥值确定的指针值,并且

将在所述确定的指针中的每一个保持的所述密钥值和所述对应位置移动至所述第一存储器;

其中所述第二类型的一个或多个处理器还被配置来:

对于所述一个或多个确定的指针中的每一个,检索由所述一个或多个确定的指针所指向的一个或多个数据值的检索数据,其中所述检索数据是在所述第一存储器中在所述对应位置处从所述第一存储器检索的;以及

使用所述一个或多个请求的所述回调函数,用所述检索数据来执行所述一个或多个请求中的每一个。

15. 如权利要求14所述的系统,其中:

所述第一类型的所述一个或多个处理器包括一个或多个中央处理器单元核心,并且

所述第二类型的所述一个或多个处理器包括一个或多个图形处理单元核心。

16. 如权利要求15所述的系统,其中所述一个或多个图形处理单元线程是基于图形处理单元核心的持久线程。

17. 如权利要求14所述的系统,其中所述第二类型的所述一个或多个处理器对所述第二存储器的访问时间小于所述第一类型的所述一个或多个处理器对所述第二存储器的访

问时间。

18. 如权利要求14所述的系统,其中所述第一类型的所述一个或多个处理器还被配置来将所述一个或多个请求移动至所述第二存储器中的进站队列。

19. 如权利要求18所述的系统,其中:

所述第二类型的所述一个或多个处理器还被配置来执行线程以检查所述第二存储器中的所述进站队列,从而确定所述请求的所述数量何时为至少所述阈值数量,并且

所述线程为持久线程。

20. 如权利要求14所述的系统,其中所述第二类型的所述一个或多个处理器还被配置来将所述检索数据移动至所述第一存储器中的出站队列。

使用异构处理器为应用程序提供低潜伏时间的系统和方法

[0001] 相关申请的交叉引用

[0002] 本申请要求2012年6月8日申请的美国临时专利申请序列号61/657,404的优先权,所述申请的完整内容特此以引用方式并入,如同完全阐明一样。

技术领域

[0003] 本发明的实施方案涉及为应用程序提供低潜伏时间(latency),并且更具体来说,涉及使用异构处理器提供低潜伏时间。

[0004] 背景

[0005] 一些计算机系统包括多于一种的处理器类型。例如,一些计算机系统包括一个或多个中央处理器单元(CPU)(即,第一处理器类型)和许多外围处理器(即,不同或第二处理器类型)。外围处理器常常为图形处理单元(GPU),但是其它处理器类型对于本领域的普通技术人员是已知的。可能存在可具有与CPU单独共享的存储器的许多GPU。一些应用程序仅使用CPU,或以不太有效的方式使用GPU。

[0006] 另外,一些应用程序需要来自计算机系统的低潜伏时间或延迟,以响应来自应用程序的请求。常常必须采购其它硬件来确保响应来自应用程序的请求的延迟不是很长。

[0007] 因此,本领域中需要使用异构处理来为应用程序提供低潜伏时间的系统和方法。

[0008] 实施方案概述

[0009] 本发明公开用于响应请求的方法、装置和计算机可读介质。一种用于响应请求的方法可包括一个或多个中央处理单元(CPU)接收一个或多个请求。所述方法可包括将一个或多个请求从与一个或多个CPU相关联的第一存储器移动至与一个或多个图形处理单元(GPU)相关联的第二存储器。所述方法可包括一个或多个GPU确定一个或多个请求中的每一个的指针。所述指针可基于请求中的信息来确定。所述方法可包括将确定的指针移动至第一存储器。对于确定的指针中的每一个,所述方法可包括检索由确定的指针所指向的数据。所述数据可从第一存储器中的第一数据结构检索。而且,所述方法可包括一个或多个CPU通过发送对应检索数据来响应接收的请求。

[0010] 在另一个实施方案中,响应请求的方法可包括接收包括回调函数的一个或多个请求。所述一个或多个请求可接收于与一个或多个CPU相关联的第一存储器中。所述方法可包括将一个或多个请求移动至第二存储器。第二存储器可与一个或多个GPU相关联。所述方法可包括一个或多个GPU线程在一个或多个请求的数量为至少阈值数量时处理一个或多个请求,以便确定一个或多个请求中的每一个的结果。所述方法可包括将结果移动至第一存储器。而且,所述方法可包括一个或多个CPU利用对应结果执行一个或多个回调函数中的每一个。

[0011] 本发明公开一种用于响应请求的系统。所述系统可包括一个或多个CPU,所述CPU被配置来接收包括回调函数的一个或多个请求。所述一个或多个请求可接收于与一个或多个CPU相关联的第一存储器中。一个或多个CPU可被配置来将一个或多个请求移动至第二存储器。第二存储器可与一个或多个GPU相关联。而且,一个或多个CPU可被配置来利用对应结

果执行一个或多个回调函数中的每一个。一个或多个GPU可被配置来执行一个或多个GPU线程,以在一个或多个请求的数量为至少阈值数量时处理一个或多个请求,从而确定一个或多个请求中的每一个的结果。而且,一个或多个GPU可被配置来将确定的结果移动至第一存储器中。

[0012] 附图简述

[0013] 本发明的实施方案在附图的各图中以举例而非限制的方式示出,附图中的相同参考数字指示类似元件,并且其中:

[0014] 图1为可实施一个或多个公开的实施方案的示例性设备的框图;

[0015] 图2示出根据一些公开的实施方案的使用异构处理器来实现低潜伏时间应用程序的系统;

[0016] 图3和图4示意性地示出可需要低潜伏时间以用于响应请求的存储器超高速缓存应用程序的操作;

[0017] 图5和图6示出根据一些公开的实施方案的对存储器超高速缓存应用程序使用异构处理器来实现低潜伏时间应用程序的系统的操作;

[0018] 图7示意性地示出根据一些公开的实施方案的使用异构处理器来实现低潜伏时间应用程序的系统的实施方案;

[0019] 图8示出根据一些公开的实施方案的GPU可运行的内核;

[0020] 图9示出用于调用根据一些公开的实施方案的用于异构处理器的低潜伏时间应用程序的系统的数据结构和调用;以及

[0021] 图10示出根据一些公开的实施方案的对存储器超高速缓存应用程序使用异构处理器来提供低潜伏时间的系统和方法的经验检验的结果表。

[0022] 详述

[0023] 图1为可实施一个或多个公开的实施方案的示例性设备100的框图。设备100可包括例如计算机、游戏设备、手持式设备、机顶盒、电视、移动电话或平板计算机。设备100包括处理器102、存储器104、存储装置106、一个或多个输入设备108和一个或多个输出设备110。设备100还可任选地包括输入驱动器112和输出驱动器114。应理解的是,设备100可包括图1中未示出的其它部件。

[0024] 处理器102可包括具有第一类型的一个或多个第一处理器(例如,中央处理单元(CPU)) 128,其可包括一个或多个核心132;以及如图形处理单元(GPU) 130的一个或多个第二类型的处理器,其可包括一个或多个计算单元(CU) 134或GPU核心。CPU 128和GPU 130可定位在同一晶粒或多个晶粒上。CU 134可组织成群组,所述群组具有控制一组CU 134的处理控件(未示出)。处理控件可控制一组CU 134,以使得一组CU 134作为单指令多数据(SIMD)处理单元(未示出)执行。CU 134可包括可与一个或多个其它CU 134共享的存储器139。例如,处理控件可控制一百三十二个CU 134,并且这一百三十二个CU 134可全部与处理控件共享同一存储器139。

[0025] 除了GPU 130和CPU 128以外,可能存在其它类型的处理器或计算元件,如数字信号处理器(DSP)、应用程序处理器等。CPU 128可包括在CPU 128的核心间共享的存储器136。在一些公开的实施方案中,存储器136为L2超高速缓存。GPU 130可包括在一个或多个GPU 130的CU 134间共享的存储器138。数据可通过存储器136与存储器138与存储器139之间的

传递137。GPU 130和CPU 128可包括其它存储器,如用于每个核心132的存储器和用于未示出的CU 134的处理单元中的每一个的存储器。存储器136、138和104可为相干超高速缓存系统(未示出)的一部分。在一些实施方案中,存储器136、138和104的一个或多个可不为相干存储器。存储器104可定位在与处理器102相同的晶粒上,或可定位成与处理器102分开。存储器104可包括易失性或非易失性存储器,例如随机存取存储器(RAM)、动态RAM(DRAM)或超高速缓存。

[0026] 存储装置106可包括固定或可移动存储装置,例如硬盘驱动器、固态驱动器、光盘或闪存驱动器。输入设备108可包括键盘、小键盘、触摸屏、触摸板、检测器、麦克风、加速计、陀螺仪、生物识别扫描器或网络连接(例如,用于发射和/或接收无线IEEE 802信号的无线局域网卡)。输出设备110可包括显示器、扬声器、打印机、触觉反馈设备、一个或多个灯、天线或网络连接(例如,用于发射和/或接收无线IEEE 802信号的无线局域网卡)。

[0027] 输入驱动器112与处理器102和输入设备108通信,并且容许处理器102接收来自输入设备108的输入。输出驱动器114与处理器102和输出设备110通信,并且容许处理器102发送输出至输出设备110。应指出的是,输入驱动器112和输出驱动器114为任选部件,并且即使输入驱动器112和输出驱动器114不存在,设备100也将以相同方式操作。

[0028] 图2示出使用异构处理器来实现低潜伏时间应用程序的系统。图2中示出CPU 128、GPU 130、CU 134、存储器138、请求202、响应204、设置220、网络线程206、主机线程208、出站队列210、CPU数据结构216、GPU线程212、进站队列214和GPU数据结构218。请求202由网络线程206接收并且传递至主机线程208,所述主机线程208将请求202放置在进站队列214中。GPU线程212使用GPU数据结构218处理进站队列214中的请求202,并且将响应204发送至出站队列210,在所述出站队列210处,主机线程208可使用CPU数据结构216处理响应204。响应204可接着被发送至网络线程206以供发送。网络线程206可接收设置220,所述设置220可用于建立或修改CPU数据结构216和GPU数据结构218。

[0029] 请求202可为从应用程序(未示出)接收的用于参考或用于处理的请求202。请求202可通过计算机网络(未示出)接收。请求202的实例可为对数据222的请求,所述数据222对应于请求202中的密钥。请求202可包括回调函数(参见图9)。

[0030] 响应204可为对请求202的响应204。示例性响应204可为对应于请求202中的密钥(参见图3)的数据222。设置220可为用于修改或建立CPU数据结构216的指令。示例性设置220可为将被嵌入至CPU数据结构216中的新数据222。

[0031] 网络线程206可被配置来从输入设备108获得请求202和设置220,并且通过输入设备108发出响应204。例如,网络线程206可为多任务操作系统中的线程,所述线程使用插口来监控用于请求202和设置220的一个或多个传输控制协议(TCP)端口,并且使用TCP通过一个或多个端口发出响应204。网络线程206可被配置来发送或传递请求202和设置220至主机线程208,并且从主机线程208接收响应204。CPU 128可执行网络线程206。在一些实施方案中,网络线程206可驻留在存储器136(参见图1)中,和或存储器104或与核心132相关联的另一个存储器(未示出)中。在一些实施方案中,网络线程206可为应用程序线程。

[0032] 主机程序208可被配置来接收请求202并且将所述请求202放置在进站队列214中,以供GPU 130进行处理。主机线程208可被配置来从出站队列210接收响应204。在一些公开的实施方案中,主机线程208可监控出站队列210,并且当一个或多个响应204变得可用时,

主机线程208可获得响应204并且根据CPU数据结构216对响应204执行进一步处理。例如,主机线程208可从进站队列214获得响应204,并且在响应204中使用指针224来从CPU数据结构216检索数据222,并且修改响应204来包括数据222。主机线程208可接着发送响应204至网络线程206。在一些实施方案中,如果响应204的数量低于阈值数量或频率,则主机线程208可满足响应204。在一些实施方案中,可存在多于一个的主机线程208。在一些实施方案中,每个出站队列210可存在一个主机线程208。在一些实施方案中,主机线程208可驻留在与CPU 128相关联的存储器中。在一些实施方案中,网络线程206可驻留在存储器136中,和或存储器104或与核心132相关联的另一个存储器(未示出)中。

[0033] 出站队列210可为放置有已被GPU线程212处理的请求202的队列。在一些实施方案中,出站队列210的数量和主机线程208的数量可成比例。在一些实施方案中,每个主机线程208可存在一个出站队列210。在一些实施方案中,出站队列210可驻留在存储器136中或CPU 128可访问的另一个存储器中。

[0034] GPU线程212可被配置来处理请求202。在一些实施方案中,GPU 130可组织成m个组,每组n个GPU线程212。一组n个GPU线程212可各自在单独的CU 134上运行。例如,n可为64并且m可为24。那么,将存在64*24个或1536个GPU线程212。可存在针对这组n个GPU线程212的每一个的进站队列214。例如,进站队列214.1可由GPU线程212.1至212.n提供服务。这组n个GPU线程212可为单指令多数据(SIMD) CU 134。这组n个GPU线程212可同时处理一组请求202。例如,如GPU线程212.1至GPU线程212.n(其中n=64)的一组n个GPU线程212可监控进站队列214.1,并且当进站队列214.1上存在64个可用请求202时,这组GPU线程212.1至GPU线程212.64可同时处理64个请求202。在一些实施方案中,这组n个GPU线程212中的GPU线程212中的一个可监控针对这组n个GPU线程212的进站队列214。GPU线程212可运行相同内核或程序,或被配置来以相同方式处理请求202。GPU线程212可发送响应204至出站队列210。

[0035] 进站队列214可为放置有请求202的一个或多个队列。进站队列214可驻留在存储器138或另一个存储器中。GPU数据结构218可为驻留在与GPU 130相关联的存储器中的数据结构218。GPU数据结构218可基于一个或多个设置220构建,并且可基于其它信息构建。GPU数据结构218可包括指针224,所述指针224可为可用于从CPU数据结构216检索数据222的指针224。GPU数据结构218可由GPU130使用来处理请求202。在一些实施方案中,GPU数据结构218可驻留在存储器138中,和或存储器104或与GPU 130相关联的另一个存储器(未示出)中。

[0036] CPU数据结构216可为驻留在与CPU 128相关联的存储器中的数据结构216。CPU数据结构216可基于一个或多个设置220构建,并且可基于其它信息构建。CPU数据结构216可包括数据222,所述数据222可为由指针224所指向的数据222。CPU数据结构216可由CPU 128使用来处理请求202。在一些实施方案中,CPU数据结构216可驻留在存储器136中,和或存储器104(参见图1)或与CPU 128相关联的另一个存储器(未示出)中。

[0037] 图3和图4示意性地示出可能需要低潜伏时间以用于响应请求的存储器超高速缓存应用程序的操作。图3和图4中示出客户端302、设置220、确认330、服务器390、散列表326以及请求202和响应204。客户端302选择服务器390并且接着向选择的服务器390.1发送设置220,所述设置220可为用于将值338.15与密钥322.15相关联的命令。存储器超高速缓存应用程序(未示出)接收设置220并且在数据结构中存储与密钥322.15相关联的值338.15,所

述数据结构可为散列表326并且可发送确认330至客户端302。客户端302可接着发送具有秘钥322.15的请求202(参见图4),并且服务器390.1通过搜索散列表326发送具有与秘钥322.15相关联的值338.15的响应204。

[0038] 客户端302可通过如LAN或互联网(未示出)的通信网络与服务器390.1通信。在一些实施方案中,客户端302可驻留在服务器390.1上。设置220可为包括秘钥322和值338的对324的命令。秘钥322和值338可为数据。秘钥322可为识别值338的唯一方式。确认330可为设置220是否成功的指示。散列表326可为将索引328与秘钥322和值338的对324相关联的表。

[0039] 客户端302可选择服务器390。在一些实施方案中,客户端302基于秘钥322选择服务器390。例如,客户端302可基于确定秘钥322的散列值(如秘钥322的模量)来确定服务器390。例如,服务器390可基于在如图3中示出存在三个服务器390时确定(秘钥322模量3)+1的值来进行选择。

[0040] 客户端302可接着发送设置220至服务器390.1。存储器超高速缓存应用程序(未示出)可确定秘钥322的索引328,在一些实施方案中这被称为确定散列值。例如,如果散列表为9个条目,则存储器超高速缓存应用程序可将索引确定为 $[\text{秘钥322模量}9]+1$,以使得具有为30的值的秘钥322将具有 $[30 \bmod 9]+1=4$ 的散列值或索引。存储器超高速缓存应用程序将接着在散列表326中存储秘钥322.15和值338.15的对324.15。索引328的每一个可具有一连串的对324,所述一连串的对324可能需要加以遍历来搜索对324。

[0041] 以这种方式,客户端302可使得服务器390.1建立存储秘钥322和值338的对324的散列表326。客户端302可通过基于如上所述的秘钥322选择服务器390来检索与秘钥322相关联的值338,并且接着发送请求202至具有秘钥322.15的服务器390.1(参见图4)。服务器390.1接着获得秘钥322.15并且计算秘钥322.15的散列值,以确定索引328.4,并且接着在与索引328.4相关联的对324中搜索具有秘钥322.15的对324.15。当找到秘钥322.15时,存储器超高速缓存应用程序检索值338.15,并且向客户端302发送具有值338.15的响应204。

[0042] 因此,客户端302可在散列表326中设置220值秘钥322和值338的对324,并且使用秘钥322从散列表326请求202值338。在一些实施方案中,散列表326可较大并且散列表326可存储在如104、136或138的随机存取存储器(参见图1)中,以使得请求202可较快速。在一些实施方案中,可为重要的是:快速地执行设置220和/或请求202命令,以使得当客户端302请求202值338时与当值338实际上在响应204中返回时之间存在低潜伏时间。例如,散列表326可用于为需要非常快速的响应204的路由选择存储网络地址。

[0043] 图5和图6示出根据一些公开的实施方案的对存储器超高速缓存应用程序使用异构处理器来实现低潜伏时间应用程序的系统的操作。图5和图6中示出客户端302、设置220、确认330、服务器390.1、CPU数据结构216、GPU数据结构218以及(参见图6)请求202和响应204。CPU数据结构216可具有地址570和值338的数据结构。值338可使用地址570来设置或检索。GPU数据结构218包括秘钥322和地址570的由索引528访问的对524。在一些实施方案中,GPU数据结构218可为散列表527。

[0044] 从客户端302的角度,存储器超高速缓存应用程序的操作与结合图3和图4的描述相同。参考图2和图5,在操作中,客户端302发送设置220至服务器390.1。服务器390.1的网络线程206发送设置220至主机线程208。主机线程208在CPU数据结构216的地址570.7处设置值338.15。主机线程208接着在入站队列214中放置秘钥322.15和地址570.7的对524.15。

在一些实施方案中,主机线程208可基于进站队列214的内容确定将对524.15放置在哪个进站队列214上。GPU线程212接着为密钥322.15确定索引528,并且以索引528将密钥322.15和地址570.7的对524.15放置在散列表527中,所述索引528为如图5中示出的索引528.4。在一些实施方案中,对524可存储在散列表527中,作为与索引528相关联的连接表。确认330可被发送至客户端302以确认设置220是成功的。

[0045] 参考图2和图6,客户端302可发送请求202至服务器390.1。网络线程206可接受请求202。网络线程206可发送请求202至主机线程208。主机线程208可将请求202放置在进站队列214中。GPU线程212可通过确定对应于密钥322的索引528处理请求202。如图6中示出,索引528.4对应于密钥322.15,并且GPU线程212可搜索列表,以找出密钥322.15和地址570.7的对524.15。GPU线程212可接着将密钥322.15和地址570.7的对524.15移动至出站队列210。主机线程208可接着使用地址570.7以从CPU数据结构216检索值338.15。主机线程208可接着向网络线程206指示具有值338.15的响应204应发送至客户端302。网络线程206可接着发送具有值338.15的响应204至客户端302。

[0046] 在一些公开的实施方案中,CPU数据结构216可驻留在存储器136中。在一些公开的实施方案中,GPU数据结构218可驻留在存储器138中。一些公开的实施方案具有的优势在于,可为大量数据的值338可不需要传递至如存储器138的存储器,这种传递是耗时的。

[0047] 在一些公开的实施方案中,可能存在比CPU 128核心132更多的请求202。在一些公开的实施方案中,许多请求202在进站队列214中排队,直到请求202的数量等于或大于GPU 130的计算单元134的数量,并且接着一个或多个请求202被分配至GPU 130的计算单元134中的每一个。

[0048] 在一些公开的实施方案中,CPU 128和GPU 130使用原子读取/写入指令。在一些公开的实施方案中,GPU 130轮询存储器位置来获得由CPU 128写入的进站队列指针。在一些实施方案中,在GPU 130上运行的线程中的一个线程可轮询存储器位置,以用于更新进站队列214。在一些公开的实施方案中,GPU 130通过将指针写入CPU 128轮询的存储器位置来更新出站队列210。

[0049] 在一些公开的实施方案中,GPU线程212可为保持活动与内核保持活动一样久的持久线程。内核可具有响应停机消息的无限外循环。在一些公开的实施方案中,GPU 130的每个计算单元134可由两个持久线程使用OpenCL。每个计算单元134两个持久线程可提供的优势在于:虽然第一线程可正在等待数据抵达,但第二线程可执行。

[0050] 图7示意性地示出使用异构处理器来实现低潜伏时间应用程序的系统的实施方案。系统700包括CPU 128、主机线程708、出站队列210、GPU 130、GPU线程212、进站队列214。应用程序线程706可发送请求202,并且系统700可处理请求202并且发送响应204至应用程序线程706。

[0051] 应用程序线程706可为在CPU 128或另一个CPU 128上运行的应用程序。请求202可为用于处理任务的请求。例如,结合图3、图4、图5和图6,设置220和请求202如结合存储器超高速缓存应用程序所公开。响应204可为对请求202的响应。例如,响应204可为如结合存储器超高速缓存应用程序所公开的响应204。

[0052] 主机线程708可为接收请求202并且发送响应204的线程。在一些实施方案中,应用程序线程706可为密码学应用程序、网络应用程序和嵌入式应用程序。

[0053] 图8示出根据一些公开的实施方案的GPU可运行的内核。GPU线程212(参见图2)可运行gpuGenericApplication内核800,所述内核800允许GPU线程212运行通用应用程序。内核800具有从804至816的无限do循环的流程控制,所述无限do循环(do loop)可被如OpenCL®中的EOS的中断信号阻断。在805处,内核800读取可能为进站队列214的_in_control_queue,并且为_in_control_queue中的第一队列202设置read_ptr。在806处,当_in_control_queue中存在请求时,内核800在while循环(while loop)中循环。在808处,内核800根据由curr_ptr所指向的请求202调用应用程序。在810处,内核800将响应放在可为出站队列210的_out_control_queue上。在812处,内核800增加curr_ptr++。如果curr_ptr指向_in_control_queue中的需要获得服务的请求202,则内核800接着循环回到806。如果curr_ptr并未指向请求202,那么内核800前往814并且更新protocol_control。内核800接着在816处检查是否在816处接收到中断信号EOS。如果接收到中断信号EOS,则内核800结束。否则,内核800循环回到804。内核800具有的优势在于,其可被调用一次并且保持持久地响应许多请求202。

[0054] 图9示出用于调用根据一些公开的实施方案的用于异构处理器的低潜伏时间应用程序的系统的数据结构和调用。称为LOLALY_REQ的结构902可由应用程序id 903和回调函数904填充(populated)。那么,lolalySendRequest 906可用于为系统700执行请求202。

[0055] 图10示出对存储器超高速缓存应用程序使用异构处理器来提供低潜伏时间的系统和方法的经验检验的结果表。表1000包括请求大小1002并且还包括两组数据,一组针对AMD®的产品APU Brazos™、HD631/430 1004,而另一组针对AMD®的产品AP Trinity™、HD7660/600。例如,在具有2048个请求1020的情况下,APU Brazos™具有为197μ秒的潜伏时间1010以及为0.19的吞吐量MR 1006和为3.23的带宽GB 1008。在具有2048个请求1020的情况下,AP Trinity™具有为140μ秒的潜伏时间1018以及为0.31的吞吐量MR 1014和为5.26的带宽GB 1016。因此,即使具有较大数量的请求,系统也提供可接受的潜伏时间。

[0056] 应了解,可能存在基于本说明书公开内容的许多变型。尽管上述特征和元件具有特定组合,但是每个特征和元件可以在独立于其它特征和元件的情况下单独使用,或者可以用于包括或不包括其它特征和元件的各种组合中。

[0057] 所提供的方法可在通用计算机、处理器或处理器核心中实施。举例来说,合适的处理器包括通用处理器、图形处理单元(GPU)、专用处理器、常规处理器、数字信号处理器(DSP)、多个微处理器、与DSP核心相关联的一个或多个微处理器、控制器、微控制器、专用集成电路(ASIC)、现场可编程门阵列(FPGA)电路、任何其它类型的集成电路(IC)和/或状态机。可以通过使用处理的硬件描述语言(HDL)指令和包括网络列表的其它中间数据(这些指令能够存储在计算机可读介质上)的结果配置制造过程来制造这些处理器。这种处理的结果可以是光罩作品,所述光罩作品接着用于半导体制造过程中,以制造实施公开的实施方案的各方面的处理器。

[0058] 本文提供的方法或流程图可以实现在并入计算机可读存储介质中的计算机程序、软件或固件中,以由通用计算机或处理器执行。在一些实施方案中,计算机可读存储介质为非暂时性计算机可读存储介质。计算机可读存储介质的实例包括只读存储器(ROM)、随机存取存储器(RAM)、寄存器、超高速缓存存储器、半导体存储设备、如内部硬盘和可移动磁盘的磁性介质、磁光介质以及如CD-ROM盘和数字多功能盘(DVD)的光学介质。

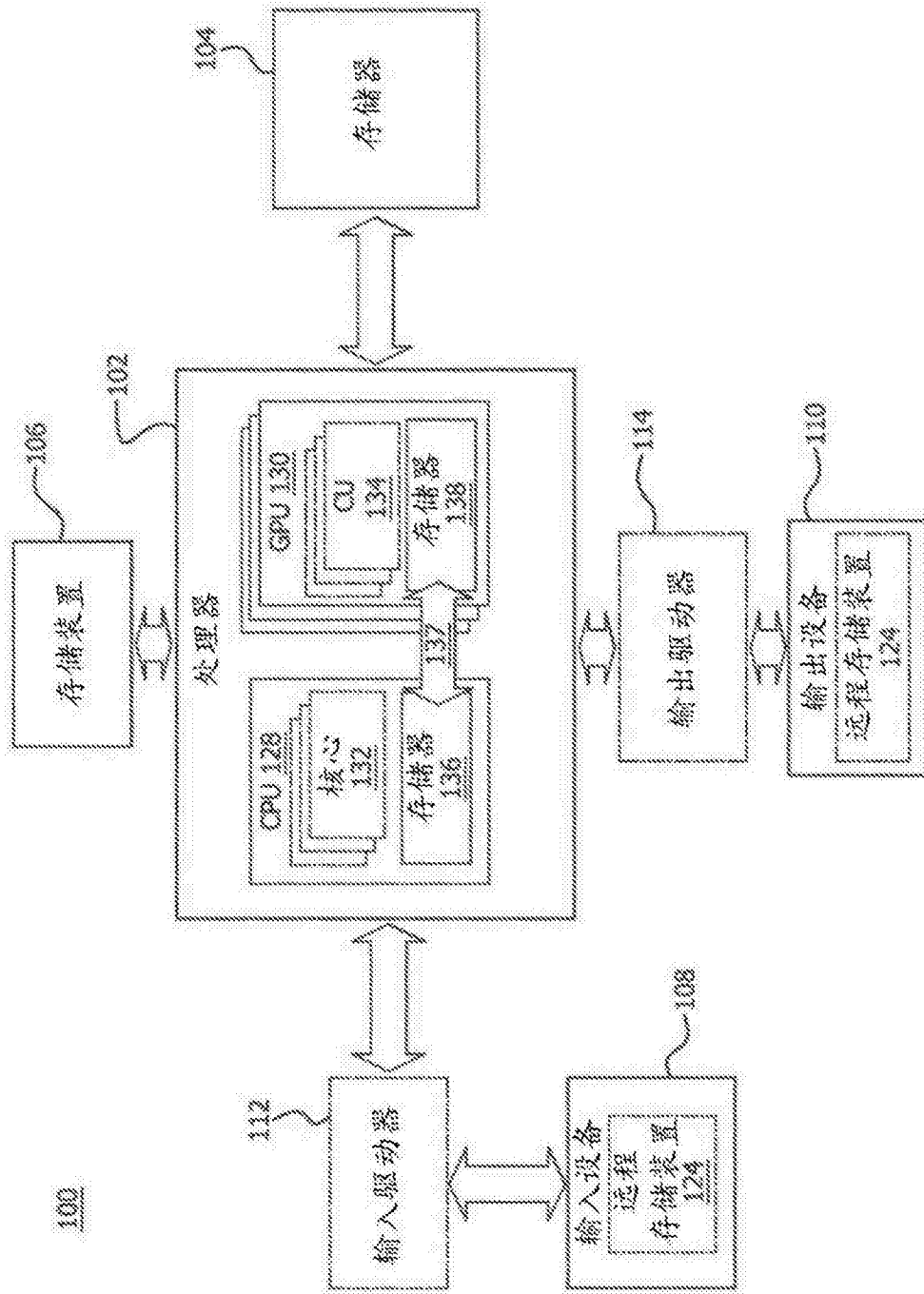


图1

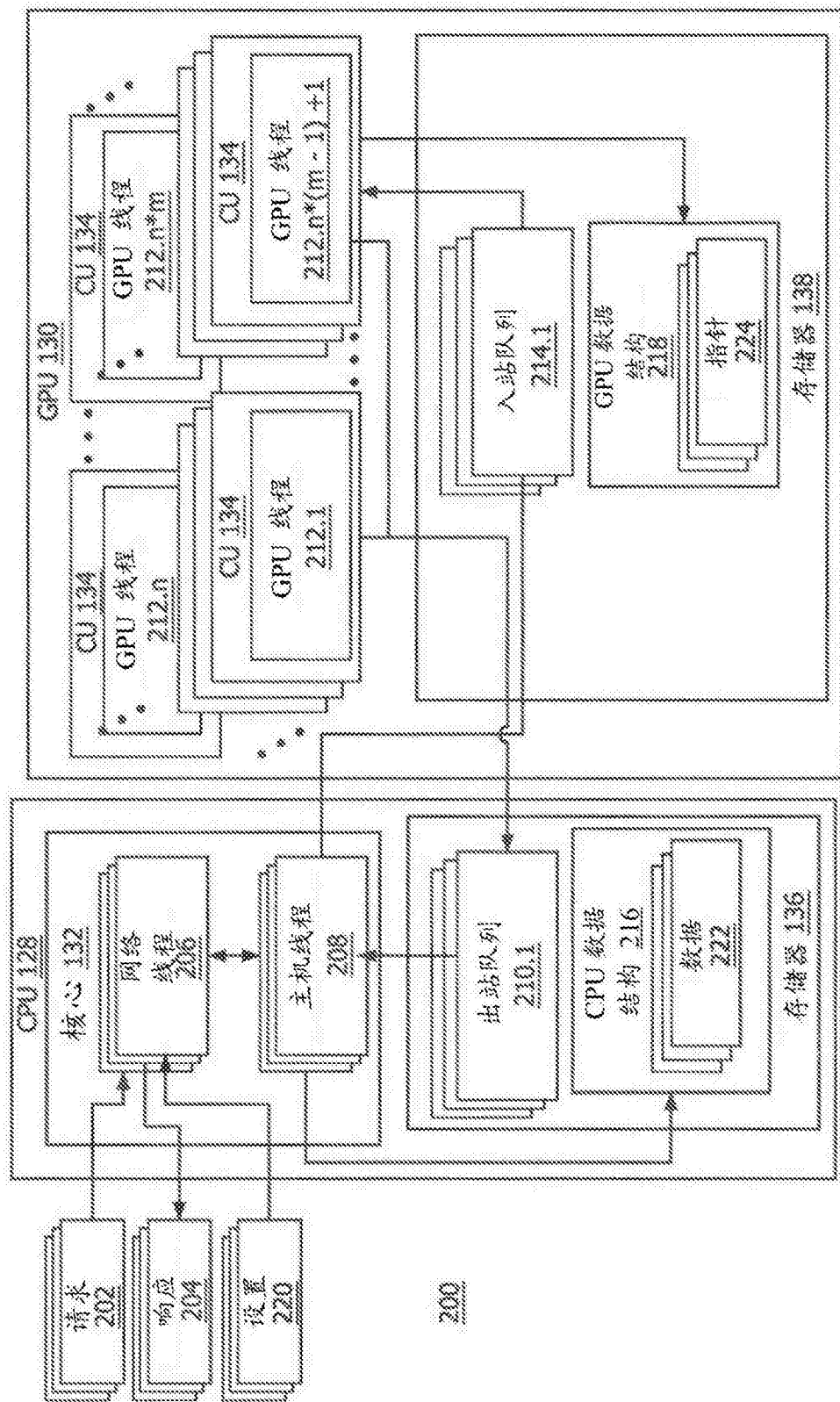


图2

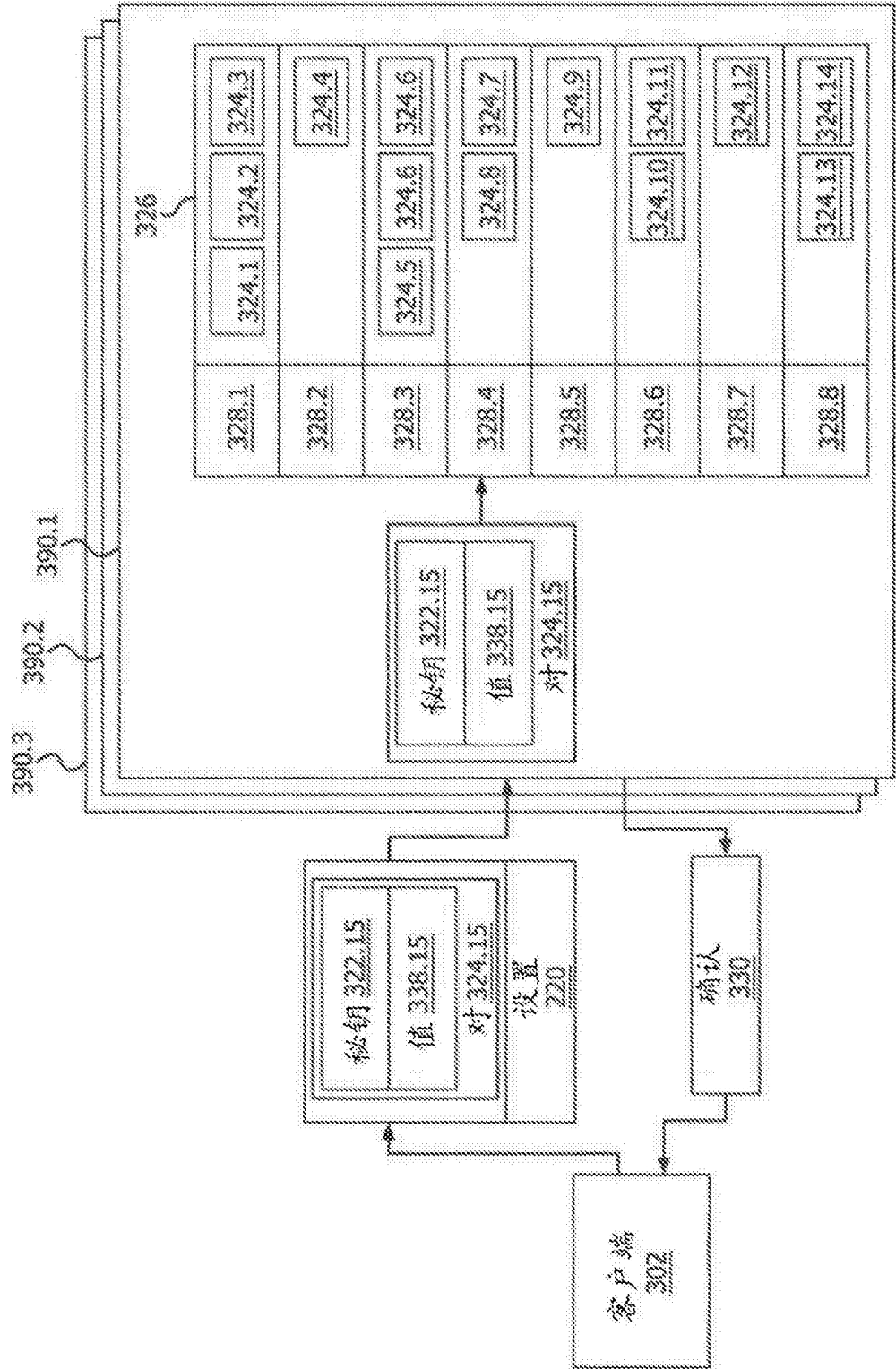


图3

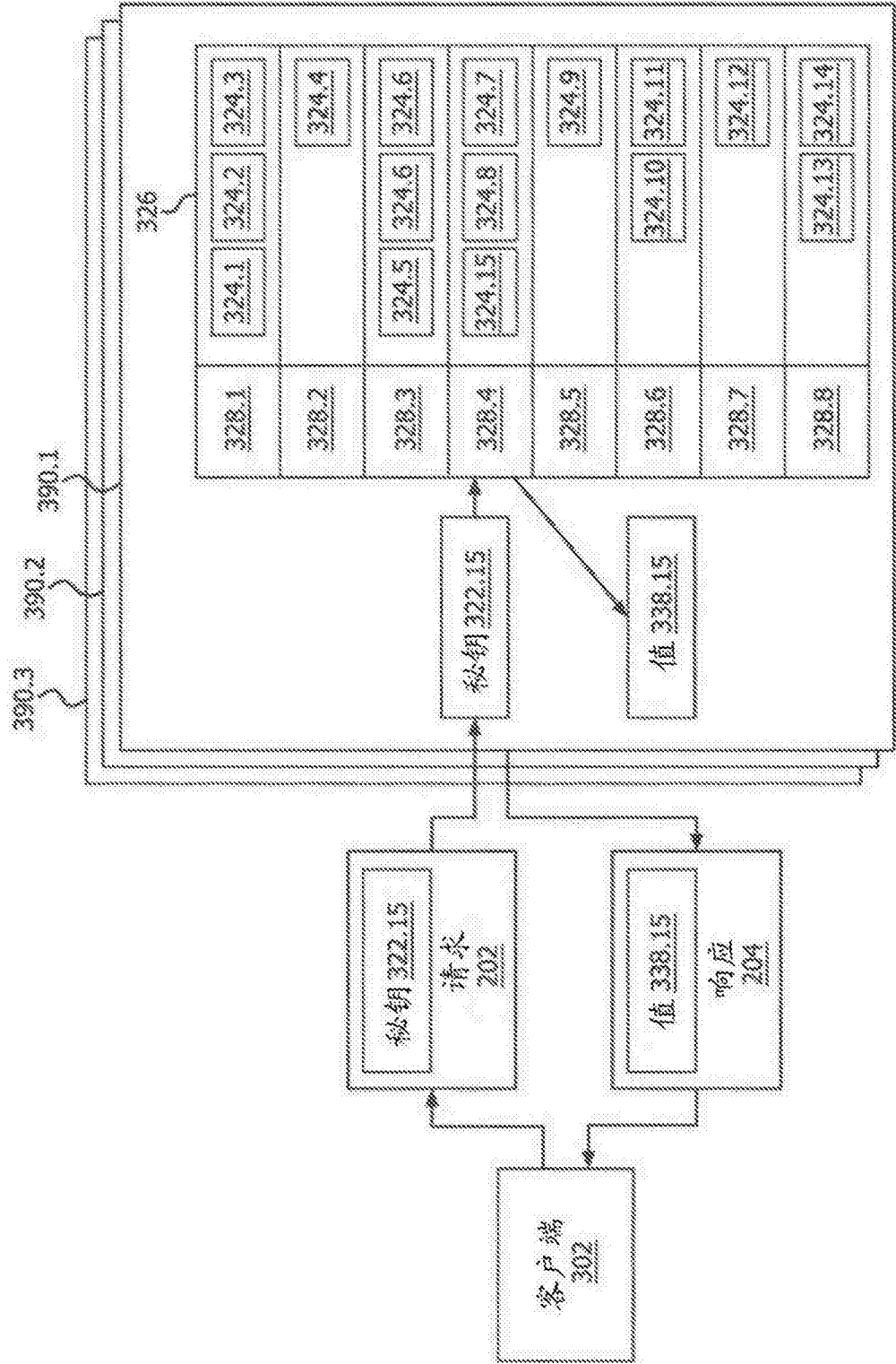


图4

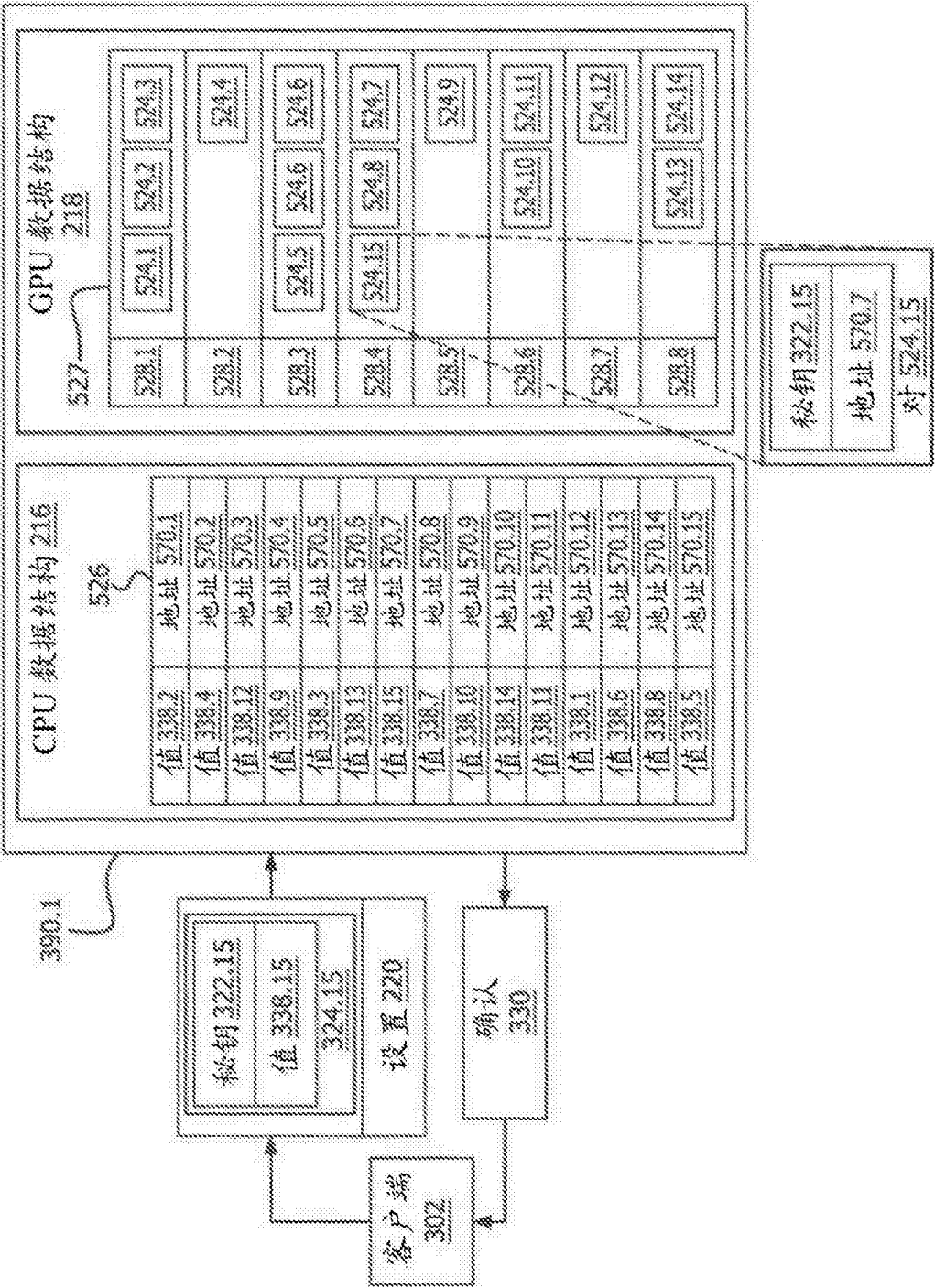


图5

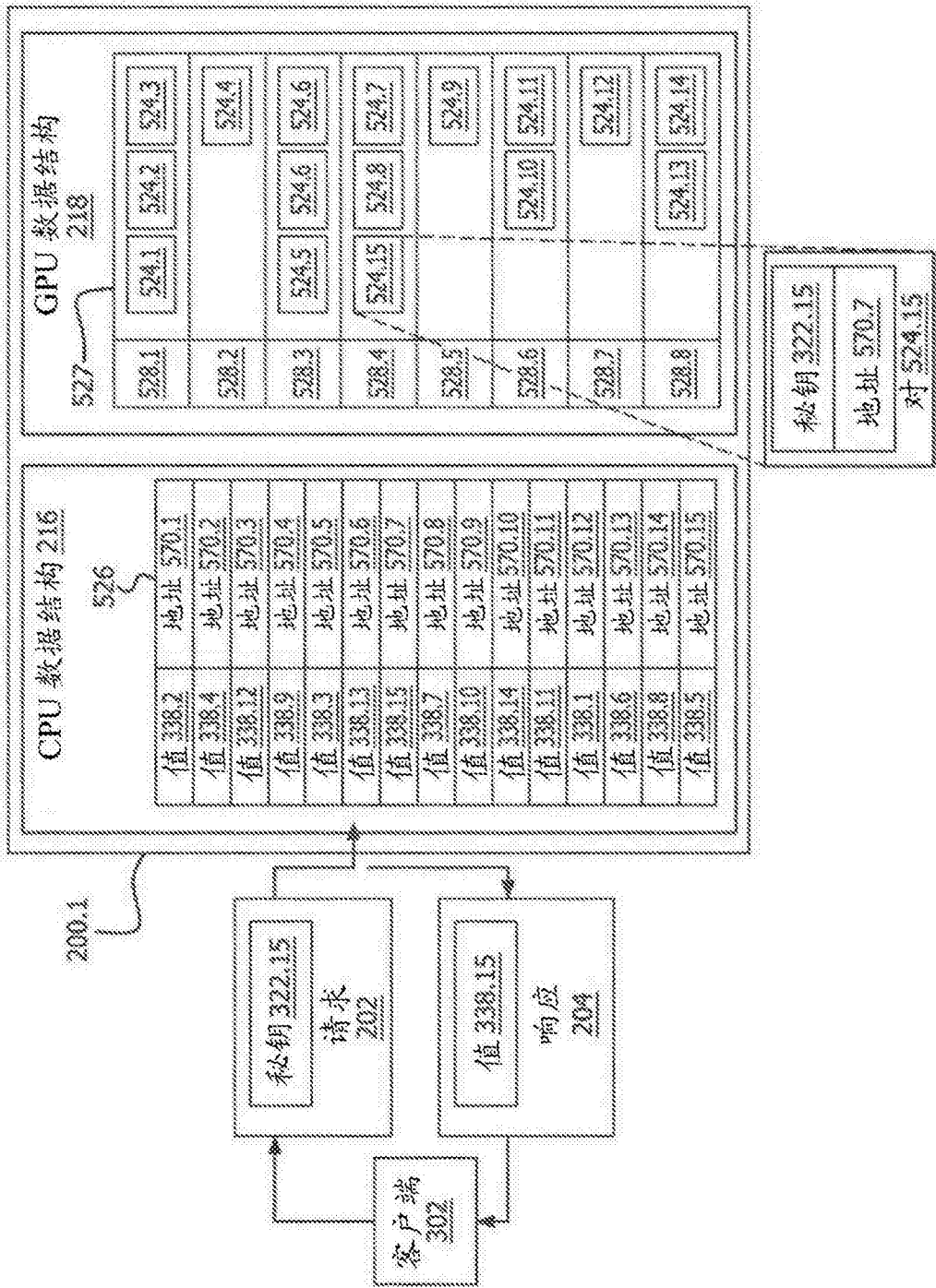


图6

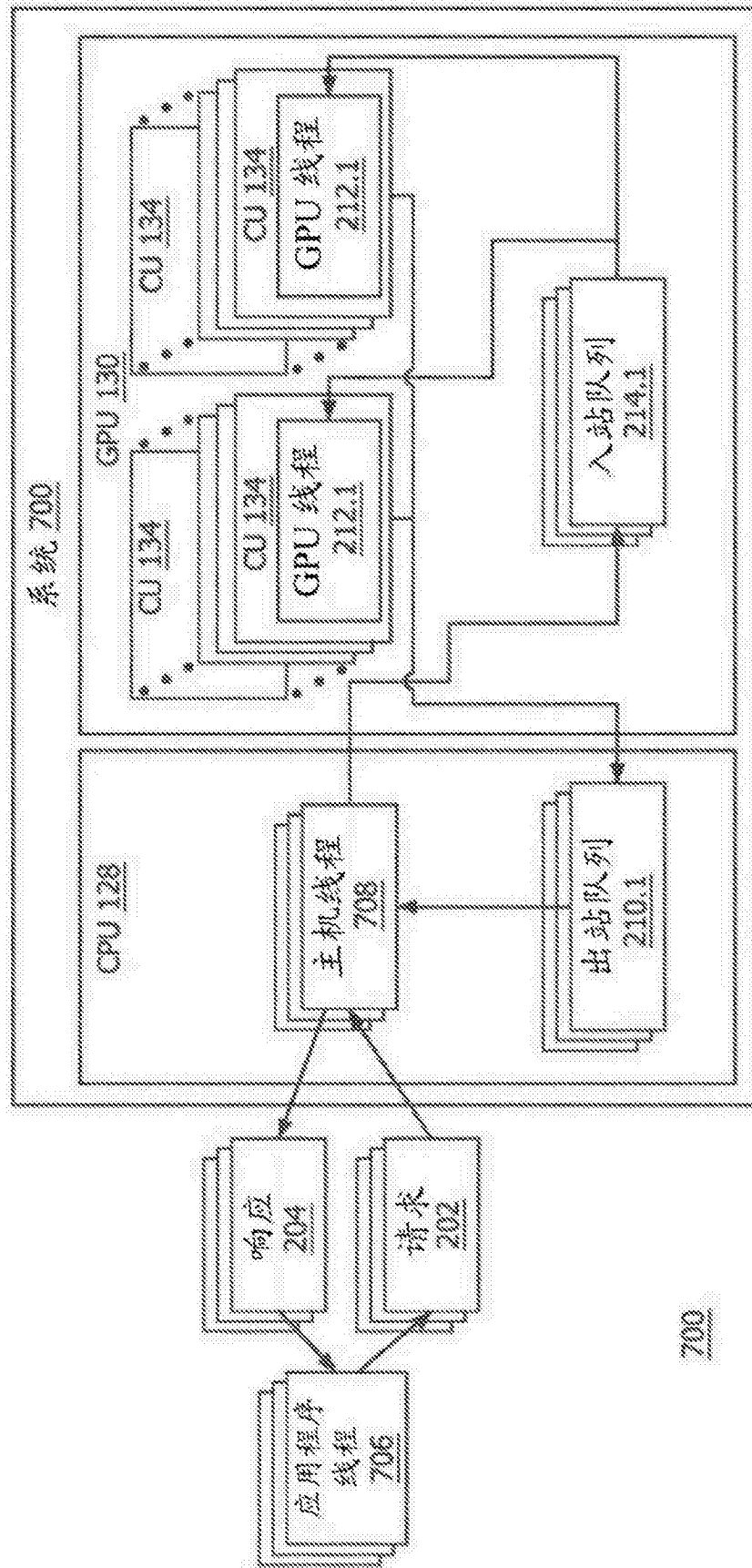


图7

800

```
802  _kernel gpuGenericApplication(_in_control_queue, _out_control_queue,  
      _app_data)  
804  {  
      curr_ptr = read_ptr = IDLE;  
      // 缓存入站队列指针  
805  read_ptr = get_read_ptr(_in_control_queue);  
      // 检查入站信号  
806  while(curr_ptr != read_ptr) {  
      // 将响应放入出站队列  
808  response = Application(_appa_data, curr_ptr);  
      // 调用应用程序  
810  set_write_ptr(_out_control_queue, curr_ptr, response);  
      // 获得下一个输入  
812  curr_ptr++;  
      }  
      // 检查EOS消息  
814  protocol_control = get_control_msg(_in_control_queue, curr_ptr);  
816  } while ( EOS != protocol_control );  
      }
```

图8

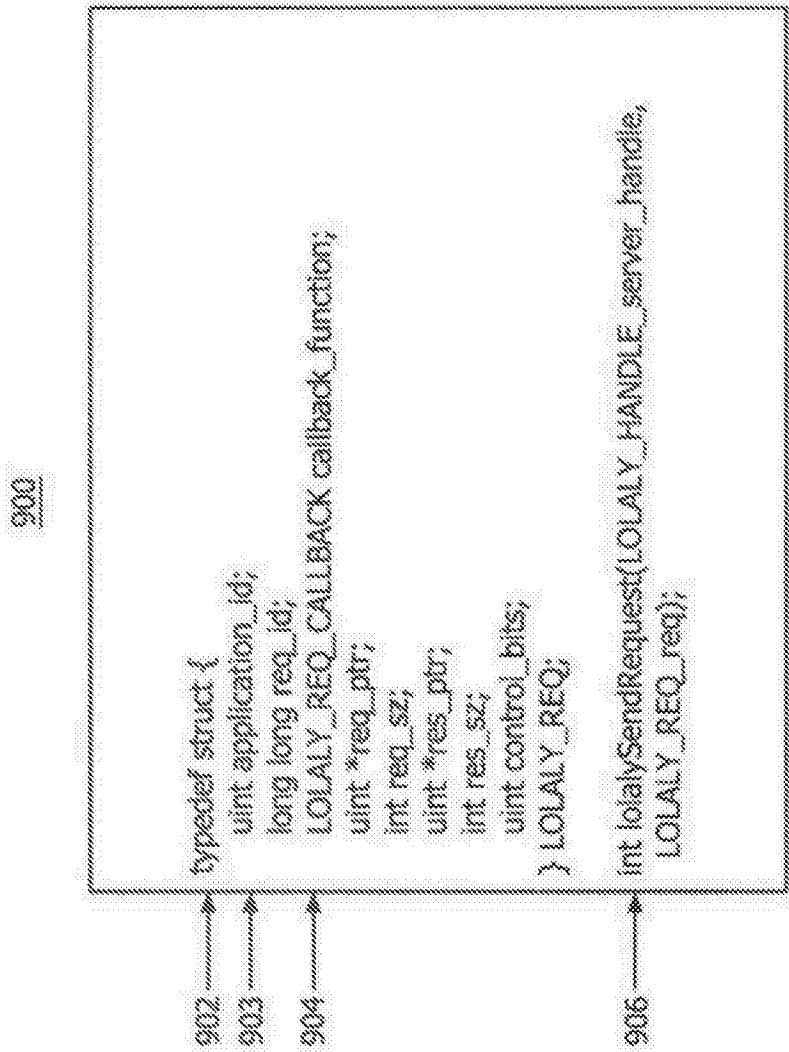


图9

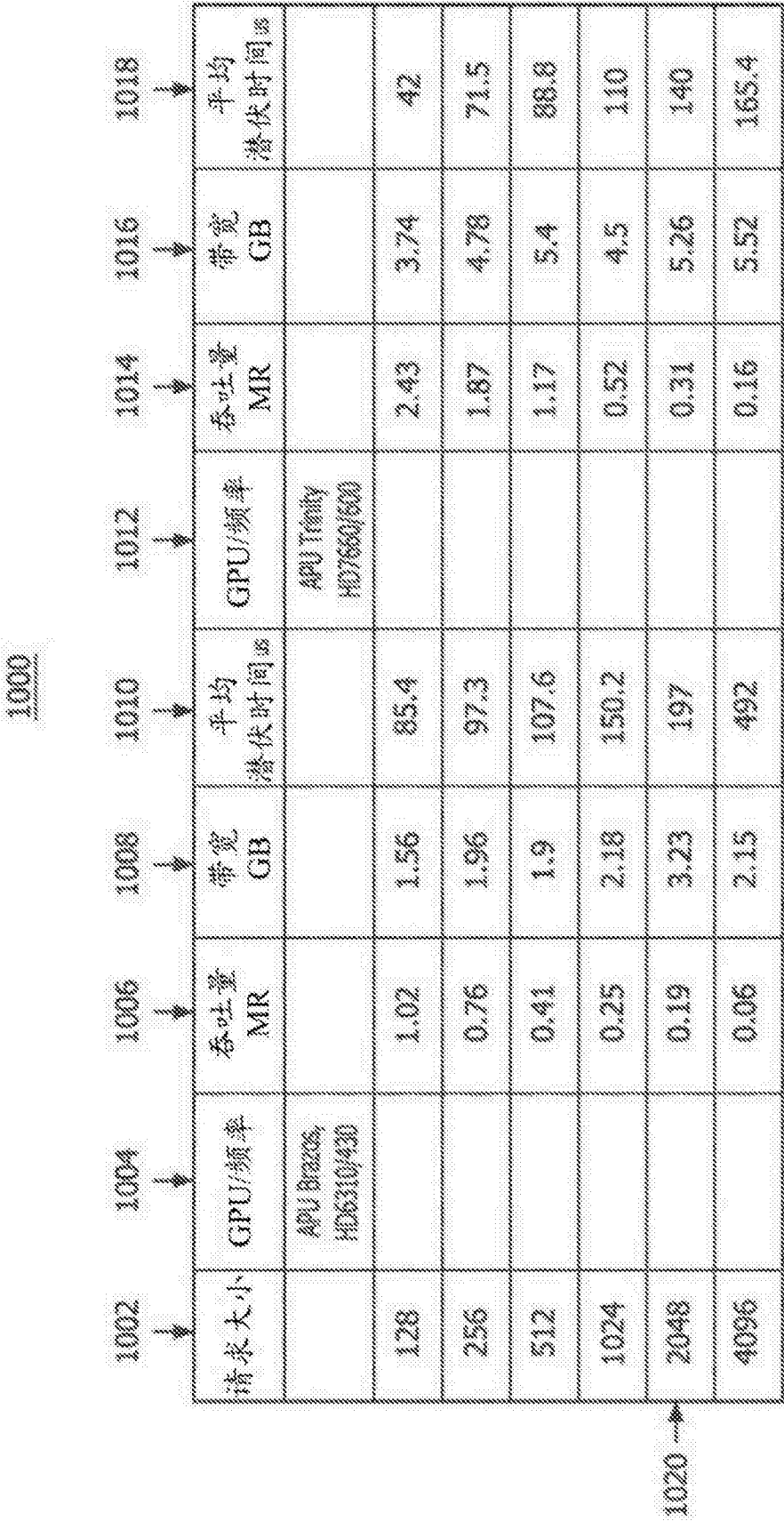


图10