(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 108009119 B
(45)授权公告日 2023.04.11

(21)申请号 201710970437.1
(22)申请日 2017.10.18
(65)同一申请的已公布的文献号
申请公布号 CN 108009119 A
(43)申请公布日 2018.05.08
(30)优先权数据
62/413,973 2016.10.27 US
62/413,977 2016.10.27 US
62/414,426 2016.10.28 US
62/485,370 2017.04.13 US
15/595,887 2017.05.15 US
(73)专利权人 三星电子株式会社
地址 韩国京畿道水原市
(72)发明人 牛迪民 李双辰 鲍勃·布伦南 克里希纳·T·马拉丁 郑宏忠
(74)专利代理机构 北京铭硕知识产权代理有限公司 11286
专利代理师 王兆赓 张川绪
(51)Int.Cl.
G06F 15/16(2006.01)
(56)对比文件
US 2009164789 A1,2009.06.25
US 2009164789 A1,2009.06.25
US 7401261 B1,2008.07.15
US 4068305 A,1978.01.10
US 2016275017 A1,2016.09.22
CN 105573959 A,2016.05.11
审查员 李文浩

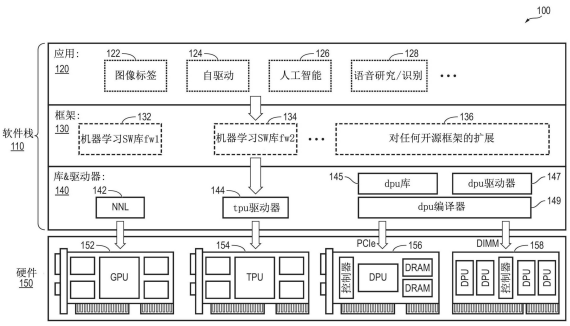
权利要求书2页 说明书11页 附图5页

(54)发明名称

处理器和控制工作流的方法

(57)摘要

公开一种处理器和控制工作流的方法。一种处理器包括:多个存储器单位,所述多个存储器单位中的每个包括多个存储器单元,其中,所述多个存储器单位中的每个可被配置为作为存储器、作为计算单元、或作为混合存储器-计算单元而操作。



CN 108009119 B

1.一种处理器,包括:

多个存储器单位,所述多个存储器单位中的每个包括多个存储器单元,

其中,所述多个存储器单位中的每个能被配置为作为存储器、作为计算单元或作为混合存储器-计算单元而操作,

其中,所述多个存储器单位中的第一存储器单位被配置为:在作为存储器、计算单元和混合存储器-计算单元中的一个而操作之后,基于存储需求和计算需求中的至少一个,被重新配置为作为存储器、计算单元和混合存储器-计算单元中的另一个而操作。

2.如权利要求1所述的处理器,其中,所述多个存储器单位中的至少一个被配置为从主机接收任务。

3.如权利要求1所述的处理器,其中,所述多个存储器单位被配置为由主机控制,主机被配置为执行所述多个存储器单位的任务划分、向所述多个存储器单位分配数据、从所述多个存储器单位收集数据和向所述多个存储器单位分配任务中的至少一个。

4.如权利要求1所述的处理器,其中,所述处理器还包括:被配置为存储存储器单位-任务映射信息的存储单元。

5.如权利要求1所述的处理器,其中,所述多个存储器单位中的每个包括DRAM。

6.如权利要求1所述的处理器,其中,所述多个存储器单位中的被配置为计算单元的存储器单位被配置为:如果所述计算单元中没有计算单元可用于执行或能够执行整个任务,则各自执行任务的对应部分。

7.如权利要求1所述的处理器,其中,所述多个存储器单位被布置为可扩展集群架构。

8.如权利要求1所述的处理器,还包括:多个存储器控制器,所述多个存储器控制器中的每个被配置为控制所述多个存储器单位中的一个或多个。

9.如权利要求8所述的处理器,还包括:用于在所述多个存储器单位之间路由工作流的多个路由器。

10.如权利要求9所述的处理器,其中,所述多个路由器中的至少一个嵌入在所述多个存储器控制器中的对应的一个中。

11.一种在包括多个存储器单位的分布式计算系统中控制工作流的方法,所述方法包括:

通过所述多个存储器单位中的一个或多个接收包括将被执行的任务的工作流,其中,所述多个存储器单位中的第一存储器单位被配置为:在作为存储器、计算单元和混合存储器-计算单元中的一个而操作之后,基于工作流的存储需求和计算需求中的至少一个,被重新配置为作为存储器、计算单元和混合存储器-计算单元中的另一个而操作;

根据工作流,通过所述多个存储器单位中的一个执行所述任务或所述任务的一部分;

在完成所述任务或所述任务的所述一部分之后,通过所述多个存储器单位中的所述一个将剩余的工作流转发给所述多个存储器单位中的另一个。

12.如权利要求11所述的方法,其中,工作流由接收任务请求的主机产生,并被提供给所述多个存储器单位中的至少一个。

13.如权利要求11所述的方法,其中,工作流由所述多个存储器单位中的一个或多个产生。

14.如权利要求11所述的方法,还包括:根据资源的可用性,将所述多个存储器单位中

的一个或多个重新配置为计算单元、存储器或混合存储器-计算单元。

15.如权利要求11所述的方法，其中，异步通信协议被用于所述多个存储器单位之间的通信。

16.如权利要求11所述的方法，其中，所述剩余的工作流被转发给所述多个存储器单位中的后续一个，直到工作流中的全部任务完成或以失败告终为止。

17.如权利要求11所述的方法，其中，如果所述多个存储器单位中的一个不能整体地完成所述任务，则所述任务被划分。

18.一种在包括多个存储器单位的分布式计算系统中控制工作流的方法，所述方法包括：

从主机接收第一请求，以通过所述多个存储器单位中的一个执行第一任务，其中，所述多个存储器单位中的所述一个被配置为：在作为存储器、计算单元和混合存储器-计算单元中的一个而操作之后，基于与第一请求对应的存储需求和计算需求中的至少一个，被重新配置为作为存储器、计算单元和混合存储器-计算单元中的另一个而操作；

通过所述多个存储器单位中的所述一个执行第一任务；

从所述多个存储器单位中的所述一个向主机提供第一任务的结果；

从主机接收第二请求，以通过所述多个存储器单位中的另一个执行第二任务。

19.如权利要求18所述的方法，其中，所述多个存储器单位中的所述另一个还从主机接收第一任务的结果。

20.如权利要求18所述的方法，其中，分布式计算系统还包括被配置为发送第一任务和第二任务并读取第一任务和第二任务的结果的主机。

处理器和控制工作流的方法

[0001] 本申请要求于2016年10月27日提交的第62/413,973号美国临时专利申请、于2016年10月27日提交的第62/413,977号美国临时专利申请、于2016年10月28日提交的第62/414,426号美国临时专利申请、于2017年4月13日提交的第62/485,370号美国临时专利申请以及于2017年5月15日提交的第15/595,887号美国专利申请的优先权权益,上述申请的全部内容通过引用包含于此。

技术领域

[0002] 根据本发明的实施例的一个或多个方面涉及一种基于DRAM的处理单元(DPU),更具体地,涉及一种DPU集群架构。

背景技术

[0003] 基于DRAM的处理单元(DPU)可用作用于其他处理器和/或图形加速器(诸如,例如,图形处理器(GPU)和专用集成电路(ASIC))的替换加速器。相应于DPU的新的生态系统可设置被设计为实现针对DPU的改进的或优化的映射和调度的驱动器和库。

[0004] DPU可以是可重新配置和可编程的。例如,DRAM单元所提供的逻辑可被配置(或重新配置)为提供不同的运算(例如,加法器、乘法器等)。例如,DPU可基于稍作修改的三晶体管一电容器(3T1C)/一晶体管一电容器(1T1C)的DRAM处理和结构。因为DPU通常不包含特定的计算逻辑(例如,加法器),因此存储器单元可用于计算。

发明内容

[0005] 本发明的实施例的多个方面指向具有多个基于DRAM的处理单元(DPU)的集群架构的方法和关联结构。

[0006] 虽然每个DPU可具有例如16千兆字节(16GB)的容量并且可在一个芯片上具有8兆(8M)计算单元,但是每个DPU可能远落后于例如包括百亿神经元的人脑。例如,可需要成百上千的DPU来实现类似人脑的神经网络(NN)。根据一个或多个示例实施例,多DPU扩展架构可用于提供类似人脑的NN。

[0007] 相比于中央处理器(CPU)/图形处理器(GPU)扩展,DPU更像是存储器(例如,DIMM)扩展,并且支持更大数量的集成。此外,通信开销可被减小或最小化。

[0008] 根据本发明的示例实施例,一种处理器包括:多个存储器单位,所述多个存储器单位中的每个包括多个存储器单元,其中,所述多个存储器单位中的每个可被配置为作为存储器、作为计算单元或作为混合存储器-计算单元而操作。

[0009] 所述多个存储器单位中的至少一个可被配置为从主机接收任务。

[0010] 所述多个存储器单位可被配置为由主机控制,主机被配置为执行所述多个存储器单位的任务划分、向所述多个存储器单位分配数据、从所述多个存储器单位收集数据或向所述多个存储器单位分配任务中的至少一个。

[0011] 处理器还可包括:被配置为存储存储器单位-任务映射信息的存储单元。

[0012] 每个存储器单位可包括DRAM。

[0013] 被配置为计算单元的存储器单位可被配置为:如果所述计算单元中没有计算单元可用于执行或能够执行整个任务,则各自执行任务的对应部分。

[0014] 所述多个存储器单位可被布置为可扩展集群架构。

[0015] 处理器还可包括:多个存储器控制器,所述多个存储器控制器中的每个被配置为控制所述多个存储器单位中的一个或多个。

[0016] 处理器还可包括:用于在所述多个存储器单位之间路由工作流的多个路由器。

[0017] 所述多个路由器中的至少一个可嵌入在所述多个存储器控制器中的对应的一个中。

[0018] 根据本发明的示例实施例,提供一种在包括多个存储器单位的分布式计算系统中控制工作流的方法。所述方法包括:通过所述多个存储器单位中的一个或多个接收包括将被执行的任务的工作流;根据工作流,通过所述多个存储器单位中的一个执行所述任务或所述任务的一部分;在完成所述任务或所述任务的所述一部分之后,通过所述多个存储器单位中的所述一个将剩余的工作流转发给所述多个存储器单位中的另一个。

[0019] 工作流可由接收任务请求的主机产生,并可被提供给所述多个存储器单位中的至少一个。

[0020] 工作流可由所述多个存储器单位中的一个或多个产生。

[0021] 所述方法还可包括:根据资源的可用性,将所述多个存储器单位中的一个或多个重新配置为计算单元或存储器。

[0022] 异步通信协议可用于所述多个存储器单位之间的通信。

[0023] 所述剩余的工作流可被转发给所述多个存储器单位中的后续一个,直到工作流中的全部任务完成或以失败告终为止。

[0024] 如果所述多个存储器单位中的一个不能整体地完成所述任务,则所述任务可被划分。

[0025] 根据本发明的示例实施例,提供一种在包括多个存储器单位的分布式计算系统中控制工作流的方法。所述方法包括:从主机接收第一请求,以通过所述多个存储器单位中的一个执行第一任务;通过所述多个存储器单位中的所述一个执行第一任务;从所述多个存储器单位中的所述一个向主机提供第一任务的结果;从主机接收第二请求,以通过所述多个存储器单位中的另一个执行第二任务。

[0026] 所述多个存储器单位中的所述另一个还可从主机接收第一任务的结果。

[0027] 分布式计算系统还可包括被配置为发送任务和读取任务的结果的主机。

附图说明

[0028] 将参照本说明书、权利要求书和附图认识并理解本发明的这些以及其他特征和方面,其中:

[0029] 图1是根据本发明的示例实施例的计算机处理架构的示意性框图;

[0030] 图2是根据本发明的示例实施例的分布式DPU集群架构的示意性框图;

[0031] 图3是根据本发明的示例实施例的具有嵌入式路由器的分布式DPU集群架构的示意性框图;

[0032] 图4是根据本发明的示例实施例的通过主机的分布式DPU集群控制的流程图，其中实现了通过主机的集中控制；

[0033] 图5是根据本发明的示例实施例的分布式DPU集群控制的流程图，其中，主机在每个计算步骤中起到积极作用；

[0034] 图6是根据本发明的示例实施例的分布式DPU集群控制的流程图，其中实现了自组织(ad hoc)控制。

具体实施方式

[0035] 本发明的实施例指向多个基于DRAM的处理单元(DPU)的方法和关联结构，其中，每个DPU被配置为DPU集群架构中的节点。根据本发明的各个实施例，每个DPU可被称为节点，或者每个DPU模块(包括多个DPU)可被称为节点。在示例实施例中，每个节点包括多个DPU模块的集合。例如，节点可包括具有多个DPU模块的服务器，其中，每个DPU模块包含多个DPU(或DPU装置)。DPU构成统一融合的能够提供正常的大规模并行处理器或处理的存储器和加速器池(memory and accelerator pool)。每个节点中的资源可受硬件(例如，算术逻辑单元(ALU)的数量等)的限制。

[0036] 根据本发明的示例实施例的计算机处理架构(或系统)可被称为包括多个存储器单位(例如，DPU)的“处理器”，其中，每个存储器单位包括多个存储器单元，所述多个存储器单元可包括3T1C存储器单元和/或1T1C存储器单元。根据示例实施例，提供灵活性以基于系统的资源需求和/或基于用户设计/喜好来配置(和/或重新配置)具有基本相同结构的存储器单位作为存储器、作为计算单元或作为混合存储器-计算单元而操作。

[0037] 图1是根据本发明的示例实施例的计算机处理架构(或系统架构)100的示意性框图。

[0038] 计算机处理架构100包括硬件(或硬件层)150，在硬件(或硬件层)150上产生软件栈来操作。计算机处理架构100可被配置用于加速深度学习，并可仿真或模拟神经网络。

[0039] 例如，硬件150包括GPU模块152、张量处理单元(TPU)模块154、基于DRAM的处理单元(DPU)模块156、以及多DPU模块158。GPU模块152和TPU模块154中的每个可包括各自的GPU或TPU以及多个支持芯片。例如，TPU可以实现在专用集成电路(ASIC)上，并可被配置或优化用于机器学习。根据示例实施例，DPU可与本领域技术人员已知的其他加速器(诸如TPU或GPU)类似地工作。

[0040] 图1中示出的DPU模块具有两种形状因子(form factor)。第一种是外围组件互连快速(PCIe)总线上的DPU模块156，第二种是双列直插存储器模块(DIMM)总线上的多DPU模块158。虽然图1中示出DPU模块156具有一个DPU装置，但是DPU模块156可以是可包括一个或多个嵌入式DPU的PCIe装置。虽然图1中示出多DPU模块158具有多个DPU装置，但是多DPU模块158可以是可包括一个或多个嵌入式DPU的DIMM。应理解，计算机处理架构100的硬件150中的DPU模块不限于PCIe装置和/或DIMM，而是可包括可包含DPU的片上系统(SoC)装置或其他存储器类型的装置。DPU的计算单元阵列可被配置为包括三晶体管一电容器(3T1C)DRAM计算单元分布(topography)和/或一晶体管一电容器(1T1C)DRAM计算单元分布。

[0041] 虽然图1中示出的硬件150具有GPU模块152、TPU模块154、DPU模块156和多DPU模块158各一个，但是在其他实施例中，硬件150可包括GPU模块、TPU模块、DPU模块和/或多DPU模

块的任何其他合适组合。例如，在一个实施例中，硬件150可仅包括DPU模块和/或多DPU模块。

[0042] 软件栈110包括一个或多个库&驱动器(例如，库&驱动器层)140、一个或多个框架(例如，框架层)130以及一个或多个应用(例如，应用层)120。该一个或多个库可包括神经网络库(NNL)142，诸如，CUDA®深度神经网络库(cuDNN)，CUDA®深度神经网络库是可从NVIDIA®得到的用于深度神经网络的图元的GPU加速库并且用于操作GPU模块152。CUDA®和NVIDIA®是加利福尼亚州圣克拉拉的NVidia公司的注册商标。当然，根据本发明的实施例，任何其他合适的商业可用和/或定制的神经网络库可代替cuDNN或额外于cuDNN使用。该一个或多个驱动器可包括用于驱动TPU模块154的TPU驱动器144。

[0043] 根据一个或多个实施例的一个或多个库&驱动器140可包括用于支持DPU硬件(例如，DPU模块156和158)的DPU库145和DPU驱动器147。DPU编译器149可用于编译使用DPU库145和DPU驱动器147创建的例程以操作DPU模块156和/或多DPU模块158。为了启用包括一个或多个DPU装置的加速器，根据示例实施例，DPU驱动器147可与TPU驱动器144非常类似。例如，DPU库145可被配置为针对可在应用层120操作的不同应用，向硬件150中的DPU中的每个子阵列提供最优映射功能、资源分配功能和调度功能。

[0044] 在一个实施例中，DPU库145可向框架层130提供可包括诸如移动、加、乘等的操作的高级应用编程接口(API)。例如，DPU库145还可包括用于标准型例程的实施方式(诸如，可应用于加速的深度学习处理的前向和后向卷积层、池化层、归一化层以及激活层，但不限于此)。在一个实施例中，DPU库145可包括映射用于卷积神经网络(CNN)的整个卷积层的计算的类似API的功能。此外，DPU库145可包括用于优化卷积层计算到DPU上的映射的类似API的功能。

[0045] DPU库145还可包括以下的类似API的功能：通过将任务内的任何单个或多个并行结构(批量、输出通道、像素、输入通道、卷积核)映射到芯片、存储体(bank)、子阵列和/或团(mat)级别的相应的DPU并行结构，来提高或优化资源分配。此外，DPU库145可包括在初始化和/或运行时提供权衡性能(即，数据移动流)和功耗的优化的DPU配置的类似API的功能。由DPU库145提供的其他类似API的功能可包括设计旋钮式(design-knob-type)功能，诸如，设置每个存储体的有源子阵列的数量、每个有源子阵列的输入特征图(feature map)的数量、特征图的划分和/或卷积核的重复使用方案。其他类似API的功能可通过向每个子阵列分配具体的任务(诸如，卷积计算、通道求和和/或数据分发)来提供额外的资源分配优化。如果操作数将在整数与随机数之间转换，则DPU库145可包括在满足精度约束的同时减小或最小化开销的类似API的功能。在精度低于期望的情况下，DPU库145可包括以下的类似API的功能：使用额外的随机表达式的位再次计算值，或者将任务卸载到其他硬件(诸如，CPU)。

[0046] DPU库145还可包括并发地(或同时地)调度DPU中的激活的子阵列并调度数据移动使得其被计算操作隐藏的类似API的功能。

[0047] DPU库145的另一方面可包括用于进一步DPU开发的扩展接口。在一个实施例中，DPU库145可提供用于使用NOR和移位逻辑直接地编程功能的接口，使得除了标准型运算(即，加、乘、最大/最小等)之外的运算可被提供。扩展接口还可提供这样的接口：使得不被DPU库145专门支持的运算可在库和驱动层140被卸载到SoC控制器、中央处理器/图形处理

器(CPU/GPU)组件和/或CPU/张量处理单元(CPU/TPU)组件。DPU库145的另一方面可提供在DPU存储器不被用于计算时将DPU的存储器用作存储器的扩展的类似API的功能。

[0048] DPU驱动器147可被配置为:提供在硬件层150的DPU、DPU库145和在更高层的操作系统(OS)之间的接口连接,以将DPU硬件层集成在系统中。也就是说,DPU驱动器147将DPU暴露于系统OS和DPU库145。在一个实施例中,DPU驱动器147可在初始化时提供DPU控制。在一个实施例中,DPU驱动器147可以以DRAM型地址或具有DRAM型地址的序列的形式,将指令发送到DPU,并且可控制数据移动到DPU内,或者移出DPU。DPU驱动器147可除处理DPU-CPU和/或DPU-GPU通信之外还提供多DPU的通信。

[0049] DPU编译器149可将来自DPU库145的DPU码编译为存储器地址的形式的DPU指令,该DPU指令被DPU驱动器147用于控制DPU。由DPU编译器149产生的DPU指令可以是对DPU中的一行和/或两行进行操作的单指令、向量指令和/或聚集向量(gathered vector)、读操作(read-on-operation)指令。

[0050] 例如,DPU模块156可将PCI快速(PCIe)接口用于通信,多DPU模块158可将双列直插存储器模块(DIMM)接口用于通信。DPU模块156除DPU之外还包括控制器和一个或多个DRAM芯片/模块。多DPU模块158可包括被配置为控制两个或更多个DPU的控制器。

[0051] 例如,多DPU模块158中的DPU可被配置为具有分布式DPU集群架构,在该分布式DPU集群架构中,DPU被布置以使得能在一个或多个DPU之间分布或共享处理或任务。例如,多DPU模块158可具有能够提供类似人脑的神经网络(NN)能力的集群架构。为提供神经网络能力,该集群架构可被配置有多个DPU、多个DPU模块(每个DPU模块包括多个DPU)、和/或多个DPU节点(每个DPU节点包括多个DPU模块)。集群架构中的每个DPU可被配置为全存储器、全计算、或组合(例如,混合存储器-计算架构)。

[0052] 框架130可包括第一机器学习软件库框架132、第二机器学习软件库框架134,并且/或者可扩展到本领域技术人员已知的用于启用DPU的一个或多个其他的开源框架136。在示例实施例中,现有的机器学习库可用于框架。例如,框架可包括Torch7和/或Tensor Flow或者本领域技术人员已知的任何其他合适的框架。

[0053] 在示例实施例中,框架层130可被配置为:将用户友好接口提供给库&驱动层140以及硬件层150。在一个实施例中,框架层130可提供用户友好接口,该用户友好接口可与在应用层120的范围广泛的应用兼容,并且使得DPU硬件层150对于用户透明。在另一实施例中,框架层130可包括将定量功能(quantitation function)添加到现有的传统方法(诸如,但不限于Torch7式应用和Tensor Flow式应用)的框架扩展。在一个实施例中,框架层130可包括将定量功能添加到训练算法的框架扩展。在另一实施例中,框架层130可将重写(override)提供给除法、乘法和平方根的现有批量标准化方法,以使现有批量标准化方法成为除法、乘法和平方根的移位近似方法。在另一实施例中,框架层130可提供允许用户设置用于计算的位的数量的扩展。在另一实施例中,框架层130可提供将来自DPU库和驱动层140的多DPU API封入(wrap)到框架层130的能力,使得用户可与使用多个GPU类似地在硬件层使用多个DPU。框架层130的另一特征可允许用户将功能分配给在硬件层150的DPU或GPU。

[0054] 在框架之上可实现一个或多个应用120,所述一个或多个应用120可包括图像标签122、自驱动算法124、人工智能126、和/或语音研究/识别128、和/或本领域技术人员已知的任何其他合适和期望的应用。

[0055] 在一些实施例中，在DPU集群架构中，主机可划分任务并针对每个划分分布/收集数据/任务。在一些实施例中，一个或多个路由器可嵌入到DIMM控制器内部，并可根据异步通信协议操作。在其他实施例中，路由器可安装在DIMM控制器或其他存储器控制器的外部(或与DIMM控制器或其他存储器控制器分开)。

[0056] 虽然主要参照DRAM(例如，3T1C DRAM或1T1C DRAM)描述了根据本发明的实施例，但是本发明不限于此。例如，在一些实施例中，任何其他合适的存储器可替代DRAM使用，以产生基于存储器的处理单元(例如，存储器单位)。

[0057] 在包括加速器的池和存储器的池的典型架构中，主机通常提供加速器与存储器之间的接口。由于主机介于加速器与存储器之间的这样的架构，所以主机可在加速器与存储器之间造成瓶颈。

[0058] 为减小或防止这样的瓶颈，在根据本发明的示例实施例中，主机不位于加速器与存储器之间。作为替代，可使用多个DPU来实现加速器。例如，每个DPU模块可包括多个DPU和DPU控制器，DPU控制器可被实现为片上系统(SoC)。此外，多个DPU模块共同连接到DPU路由器。DPU路由器可作为DPU控制器而被实现在同一SoC中。然而，本发明不限于此，可在包括DPU控制器的SoC外部实现控制器&路由器。此外，每个DPU模块可包括DPU路由器，或者单个DPU路由器可由两个或更多个DPU模块共享。

[0059] 图2是根据本发明的示例实施例的分布式DPU集群架构200的示意性框图。

[0060] 在图2的分布式DPU集群架构200中，多个DPU模块202、204、208和DRAM模块206通过控制器&路由器210、212、214、216彼此连接。虽然仅DRAM模块206被示出为被配置为存储器，但是本发明不限于此，DPU/DRAM模块中的任何模块可被配置为存储器、计算单元(或处理单元/处理器)、或混合存储器-计算单元。虽然DPU和DRAM可被分别称为加速器和存储器模块(存储器)，但是它们具有彼此基本相同的硬件结构，并且DPU和DRAM可被视为分别针对计算和存储而不同配置的DPU(或存储器单位)。此外，根据示例实施例，每个DPU可被重新配置为起加速器(用于计算)或存储器(用于存储)的作用，或者具有加速器和存储器二者的功能。

[0061] 主机220也通过控制器&路由器之一214连接到DPU。架构200可被称为以主机为中心的架构，在以主机为中心的架构中，所有的工作(例如，任务)由主机220产生。这里，主机220会知晓网络上的资源是什么，并且将通过一个或多个控制器&路由器210、212、214、216将特定命令和工作负荷发送到特定DPU。在示例实施例中，因为主机负责执行与调度和映射有关的所有任务，而DPU仅执行计算和存储，因此性能可能会被主机220约束。

[0062] 例如，当多个DPU位于一个服务器/计算机/节点的内部时，这些DPU能够彼此直接通信。对于不在同一服务器/计算机/节点中的DPU，它们可通过一个或多个路由器和/或交换机(例如，控制器&路由器210、212、214、216)彼此通信，所述通信可通过一个或多个通信路径(诸如，互联网)发生。

[0063] 每个DPU模块(例如，DPU模块202-1)包括连接到同一DPU控制器(例如，控制器SoC)的多个DPU。同一DPU模块上的DPU通过DIMM内连接(intra-DIMM connection)而连接到DPU控制器，所述DIMM内连接可以是基于总线的连接(例如，基于分层总线的连接)。因此，同一DPU模块上的这些DPU可使用安装在该同一DPU模块上的作为其所控制的DPU的DIMM上的控制器SoC来控制。这里，DPU模块中的控制器SoC可负责接收命令/数据并管理DPU模块中的DPU。

[0064] DPU模块可通过DIMM间连接彼此连接，其中，DPU模块连接到存储器控制器(未示出)，存储器控制器连接到路由器。路由器可将DPU/DRAM连接到存储器/加速器网络中。

[0065] 在图2的分布式DPU集群架构中，例如，可通过提供统一融合的存储器和加速器池来避免存储器与加速器之间的接口处的瓶颈。例如，当主机220在网络的边缘连接到控制器&路由器214时，在存储器与加速器之间很少或不存在由主机220造成的瓶颈。例如，存储器和加速器的池具有灵活的网络连接。此外，每个节点可被配置为加速器(DPU)或存储器(DRAM)。在一些实施例中，每个节点可作为包括加速器和存储器二者的特征的加速器-存储器混合体来操作。例如，每个节点可以是多个DPU模块的集合，并且可包括具有多个DPU模块的服务器，其中，每个DPU模块可包含多个DPU装置。

[0066] 因此，在根据本发明的一个或多个示例实施例的分布式DPU集群架构中，提供统一融合的存储器和加速器池以产生正常的大规模并行处理器。这里，每个节点中的资源可受硬件(例如，算术逻辑单元(ALU)的数量等)限制。换言之，受限的ALU的数量可确定DPU能够提供的最大存储器容量或最大计算能力。

[0067] DPU集群架构提供可重新配置的存储器/计算资源，并且每个DPU中的所有资源能够被配置为全存储器、全计算单元、或存储器与计算单元的组合(或混合体)。这样，根据每个节点处的存储和/或计算需求，可防止或降低存储和/或计算资源的浪费。这是因为每个节点在使用期间可根据需要或按用户所期望的那样被配置或重新配置，以提供更多存储资源或更多计算资源。

[0068] 图3是根据本发明的示例实施例的具有嵌入式路由器的分布式DPU集群架构300的示意性框图。图3的架构300与图2的架构200不同之处在于路由器被组合到DPU控制器中。分布式DPU集群架构300还可被称为以主机为中心的架构。

[0069] 与图2的DPU集群架构类似，在图3中，由于不存在位于加速器池与存储器池之间的接口的主机，因此可减小或避免加速器池与存储器池之间的接口的瓶颈。根据图3的分布式DPU集群架构300，多个DPU模块/DRAM模块302、304、306、312、314、316、322、324、326、332、334、336以行和列来布置。由于DPU模块之一312在网络的边缘连接到主机350，因此很少或不存在由主机350造成的瓶颈。

[0070] 在根据本发明的示例实施例的以主机为中心的架构中，起初主机将针对不同DPU产生工作或工作部分。例如，工作1(或工作部分1)可被分配给DPU1，工作2(或工作部分2)可被分配给DPU2。然后，主机可将工作流中的工作1发送到DPU1。工作流还可包括将发送到DPU2的工作2。当DPU1结束其工作时，由于DPU1知晓下一步是哪里(如主机所映射的/调度的)，因此DPU1可在不将结果发送回主机的情况下直接将结果发送到DPU2。例如，当DPU1结束其工作时，它可直接将中间数据(或中间结果)发送到DPU2，以用于其他计算。因此，DPU1不必将中间数据发送回主机来用于主机将接收的中间数据发送到DPU2。因此，在图3的DPU集群架构300中，DPU节点可在没有主机的情况下彼此通信。例如，主机可仅发送用于请求DPU1将数据转发到DPU2的命令，并且控制器SoC负责移动所述数据。因为主机知晓网络中的所有资源，并且主机负责执行映射和调度，所以这在根据示例实施例的图2和图3的以主机为中心的架构中是可能的。因此，主机可不必参与每一步的计算。根据本发明的一些其他示例实施例，提供如下的一种系统架构：网络上的装置不必知晓网络是什么样或网络上什么资源是可用的。

[0071] 如图3中可见，仅一个模块(即，DRAM模块324)被配置为存储器，而剩余模块被配置为处理/计算模块。在其他实施例中，一个或多个模块可被配置为存储器模块或混合计算/存储器模块。DPU/DRAM模块中的每个包括多个DPU和DPU控制器(即，控制器SoC)，使得每个模块能被配置或重新配置为DPU模块或DRAM模块。此外，即使所有DPU/DRAM具有彼此基本相同的硬件结构，但在同一DPU/DRAM模块内，一个或多个DPU可被配置为存储器，而一个或多个其他DPU可被配置为计算单元(或处理单元)。

[0072] 例如，在一些实施例中，使用DIMM控制器SoC实现路由器。此外，可使用异步通信协议来保证可使用握手的适合的DIMM间通信。在其他实施例中，可使用同步协议，诸如双数据速率(DDR)。因为主机350位于网络的一个边缘而非DPU/DRAM模块之间，所以可导致更少的主机带宽使用并且可减小或消除由主机造成的任何瓶颈。

[0073] 在一些示例实施例中，虽然DPU/DRAM或DPU模块/DRAM模块被布置为分布式DPU集群架构，但是它们仍可被主机集中控制。在这样的实施例中，主机保持DPU/DRAM任务映射信息。任务映射信息可以是软件和/或驱动程序的形式。在网络中，神经网络参数和其他有用数据可被存储在可在集群中的一个节点中的DRAM中。

[0074] 根据本发明的示例实施例，提供了两个分开的以主机为中心的架构/配置。在以第一主机为中心的架构中，主机将把工作流中的所有工作负荷转发给第一DPU，第一DPU将把剩下的工作负荷和/或结果转发给第二DPU。第二DPU将执行计算并将把结果和工作流中剩下的工作负荷转发给第三DPU，以此类推。在以第二主机为中心的架构中，在每一步，主机读取回数据，产生接下来的工作，并将接下来的工作与中间数据(例如，在先前DPU中执行的计算的结果)一起发送到接下来的DPU。在其他示例实施例中，可提供自组织(ad hoc)控制，在自组织控制中，一个或多个DPU能够在没有主机的映射/调度的情况下产生包括任务的工作流，甚至可不需要主机。可使用图1、图2和图3中的任何图中示出的合适的系统/硬件架构来实现上面的示例实施例中的每个。

[0075] 图4是根据本发明的示例实施例的通过主机的分布式DPU集群控制的流程图，其中实现了通过主机的集中控制。

[0076] 在图4的框400中，主机接收任务请求。在框402中，主机检查存储在存储器中的DPU/DRAM任务映射表。在任务映射表检查处理中，主机查找存储任务的参数的DRAM。例如，对于神经网络应用，主机可查找具有NN参数的DRAM。主机还查找可用的DPU资源。

[0077] 在框404中，如果DPU不能单独完成任务，则主机将任务划分到两个或更多个DPU资源。例如，任务可被划分成任务1、任务2等，它们可被分配到两个或更多个不同的DPU，例如，任务1被分配到DPU1，任务2被分配到DPU2等等。

[0078] 在框406中，主机产生工作流，例如，主机针对每个任务和/或任务的每个划分的部分分配资源号码和任务号码(例如，(资源#,任务#))。例如，工作流(WF)可具有以下格式:WF＝[(资源0,任务0)，(资源1,任务1)，(资源2,任务2)……(资源N,任务N)，(主机,完成)]等。这里，资源可参照DPU/DRAM(或DPU/DRAM模块)。在根据本发明的示例实施例中，工作流可具有本领域技术人员已知的任何其他合适的格式。

[0079] 在框408中，主机将工作流发送到连接到主机的DPU/DRAM。在框410中，DPU/DRAM(或DPU/DRAM模块)读取工作流包。如果工作流顶端上的“资源”与当前资源匹配，则DPU/DRAM(或DPU/DRAM模块)执行工作流顶端上的“任务”，然后从工作流移除顶端上的(资源#,

任务#)对。之后,剩余的工作流被转发给一个或多个其他资源(例如,DPU)。如果没有匹配,则DPU/DRAM将工作流转发到可以是接下来的(资源#,任务#)对中的资源的资源1。然而,本发明不限于任何特定方案,本领域技术人员已知的任何合适的方案可被使用。一旦完成工作流,则主机确定所有任务完成,这意味着所有的(资源#,任务#)对从0至N已结束。

[0080] 图5是根据本发明的示例实施例的分布式DPU集群控制的流程图,其中,主机在每个计算步骤中起到积极作用。

[0081] 在图5的框500中,主机将工作请求发送到DPU,然后在框502中,主机从DPU读取计算的结果,或DPU可将结果发送回主机。然后在框504中,主机将工作请求和/或来自DPU的结果发送到接下来的DPU,并在框506中从接下来的DPU读取计算的结果,以此类推,直到在框508中所有工作被完成为止。在这个以主机为中心的架构中,主机参与每个计算步骤中,并且更主动地控制DPU之间的工作流和数据的流动。

[0082] 与在图4和图5的流程图中描述的以主机为中心的架构的集中控制不同,根据本发明的一个或多个示例实施例,在不集中控制或者较小或最小集中控制的情况下以自组织方式控制分布式DPU集群架构。图6是根据本发明的示例实施例的分布式DPU集群控制的流程图,其中实现了自组织控制。例如,在根据一些示例实施例的自组织控制机制中,如图6的框600所示,每个DPU可产生任务(或任务的工作流)。此外,可不需要资源表,并且可不需要将DPU/DRAM任务映射信息存储在存储器中。根据这个自组织方案,如框602所示,产生任务(或任务的工作流)的DPU(例如,DPU0)完成任务或任务的一部分,然后如框604所示地使用路由信息将剩余的任务和/或任务的部分(例如,剩余的工作流)发送到一个或多个邻近DPU节点(例如,接下来的DPU或DPU1、DPU2、DPU3等)。然后,在框606中,接下来的DPU完成任务和/或任务的部分。完成任务(或其部分)以及发送剩余的任务(或其部分)的过程如框608所示地被重复,直到所有任务完成或者任务或任务分配失败(例如,不再有可用资源)为止。

[0083] 自组织控制的一些特征是:不需要主机服务器、不需要保持很大的集群信息、以及能够支持庞大集群(例如,可能多达用于产生与人脑的规模类似的神经网络的现有网络中可用的DPU的100至1000倍)。除非控制是集中式的,否则,资源管理可能低于最优,并且最终可能发生失败。失败的显著性可很大程度上依赖于应用。对于一些应用,失败可能是严重的,而在其他应用中可能不那么严重。例如,在模拟人脑行为的人工智能(AI)应用中,AI应用可记得或不记得某事。此外,AI应用可在某时记得某事而可在另一时刻不记得该事。如果根据本发明的实施例的DPU系统架构被用于大规模神经网络应用,则一些失败可以是可接受的。

[0084] 在其他实施例中,分布式DPU集群架构可使用混合集中的自组织控制来操作。例如,一些DPU和/或DPU模块可被主机控制而其他DPU和/或DPU模块可以以自组织方式控制。再如,DPU和/或DPU模块中的至少一些可被重新配置,以使得它们的控制可如期望或需要的那样在集中控制和自组织控制之间来回切换。

[0085] 因此,本发明的实施例指向布置多个基于DRAM的处理单元(DPU)并具有可提供类似人脑的神经网络能力的分布式架构的集群架构。DPU集群提供可重新配置的存储器/计算资源,使得DPU节点中的所有资源能被配置为全存储器、全计算、或组合(即,混合存储器-计算单元)。

[0086] 在具有以主机为中心的架构的一些实施例中,主机可划分任务和/或针对每个划

分分布/收集数据/任务。这里，主机可针对DPU产生并发送工作流，以完成任务并将剩余的任务和/或结果数据转发到其他DPU，或者主机可通过接收结果数据并将结果数据与任务一同转发到其他DPU来控制每个计算步骤。在其他实施例中，集群架构的控制可被设置为自组织的，其中，一个或多个DPU可产生任务/工作流，完成任务，和/或将剩余的任务/工作流发送到网络中的其他DPU。在其他实施例中，集中控制和自组织控制的混合体可被实现，以控制DPU/DRAM集群架构。路由器可根据异步通信协议而操作，并可被嵌入到DIMM控制器内部。

[0087] 将理解，虽然术语“第一”、“第二”、“第三”等可在这里用于描述各种元件、组件、区域、层和/或部分，但是这些元件、组件、区域、层和/或部分不应被这些术语所限制。这些术语用于将一个元件、组件、区域、层或部分与另一元件、组件、区域、层或部分区分。因此，在不脱离本发明的精神和范围的情况下，以上讨论的第一元件、组件、区域、层或部分可被称为第二元件、组件、区域、层或部分。

[0088] 例如，在这里描述的根据本发明的实施例的相关装置或组件可利用任何合适的硬件(例如，专用集成电路)、固件(例如，DSP或FPGA)、软件或者软件、固件和硬件的合适组合来实现。例如，相关装置的各种组件可形成于一个集成电路(IC)芯片上或分开的IC芯片上。此外，相关装置的各种组件可被实现在软性基板电路(flexible printed circuit film)、带载封装(TCP)、印制电路板(PCB)上，或者作为一个或多个电路和/或其他装置而在同一基底上被形成。此外，相关装置的各种组件可以是一个或多个计算装置中的运行在一个或多个处理器上的进程或线程，其中，一个或多个计算装置执行计算机程序指令并与用于执行这里描述的各种功能的其他系统组件交互。计算机程序指令被存储在存储器中，所述存储器可使用标准存储器装置(诸如，随机存取存储器(RAM))被实现在计算装置中。计算机程序指令还可被存储在其他非暂时性计算机可读介质(诸如，CD-ROM、闪存驱动等)中。此外，本领域技术人员应该认识到，在不脱离本发明的示例实施例的精神和范围的情况下，各种计算装置的功能可被组合或集成到单个计算装置中，或者特定计算装置的功能可被分布于一个或多个其他计算装置。

[0089] 此外，还将理解，当一个元件、组件、区域、层和/或部分被称为在两个元件、组件、区域、层和/或部分“之间”时，它可以是这两个元件、组件、区域、层和/或部分之间的仅有元件、组件、区域、层和/或部分，或者也可存在一个或多个中间元件、组件、区域、层和/或部分。

[0090] 这里使用的术语用于描述特定实施例的目的，而不意图限制本发明。如这里使用的，除非上下文清楚地另有指示，否则单数形式也意图包括复数形式。还将理解，当在本说明书中使用术语“包括”、“包含”时，指定存在陈述的特征、整体、步骤、操作、元件和/或组件，但不排除存在或添加一个或多个其他特征、整体、步骤、操作、元件、组件和/或它们的组。

[0091] 如这里使用的，术语“和/或”包括一个或多个相关所列项的任何组合和所有组合。诸如“……中的至少一个”、“……中的一个”和“从……选择”的表述当位于一列元素之后时，修饰整列元素而不修饰该列的单个元素。此外，当使用“可”描述本发明的实施例时，表示“本发明的一个或多个实施例”。

[0092] 如这里使用的，可认为术语“使用”等同于术语“利用”。

[0093] 关于本发明的一个或多个实施例描述的特征可用于结合本发明的其他实施例的

特征使用。例如，在第一实施例中描述的特征可与第二实施例中描述的特征组合以形成第三实施例，即使可能未在这里具体描述第三实施例。

[0094] 本领域的技术人员还应该认识到，处理可通过硬件、固件(例如通过ASIC)或者以软件、固件和/或硬件的任何组合来执行。此外，处理的步骤的顺序不是固定的，而是可如本领域的技术人员所认识到的那样被改变成任何期望的顺序。改变的顺序可包括所有步骤或一部分步骤。

[0095] 虽然已经关于特定具体实施例描述了本发明，但是本领域技术人员将毫不费力地设计描述的实施例的变化，所述变化不脱离本发明的范围和精神。此外，对于各种领域的技术人员而言，这里描述的本发明本身将提出其他任务的解决方案和其他应用的改编。本申请人意图通过权利要求涵盖本发明的所有这样的使用以及在不脱离本发明的精神和范围的情况下出于公开的目的而选择的在此对本发明的实施例做出的改变和修改。因此，本发明的现有实施例应在所有方面被理解为说明性的而非限制性的，并且本发明的范围由权利要求及其等同物指示。

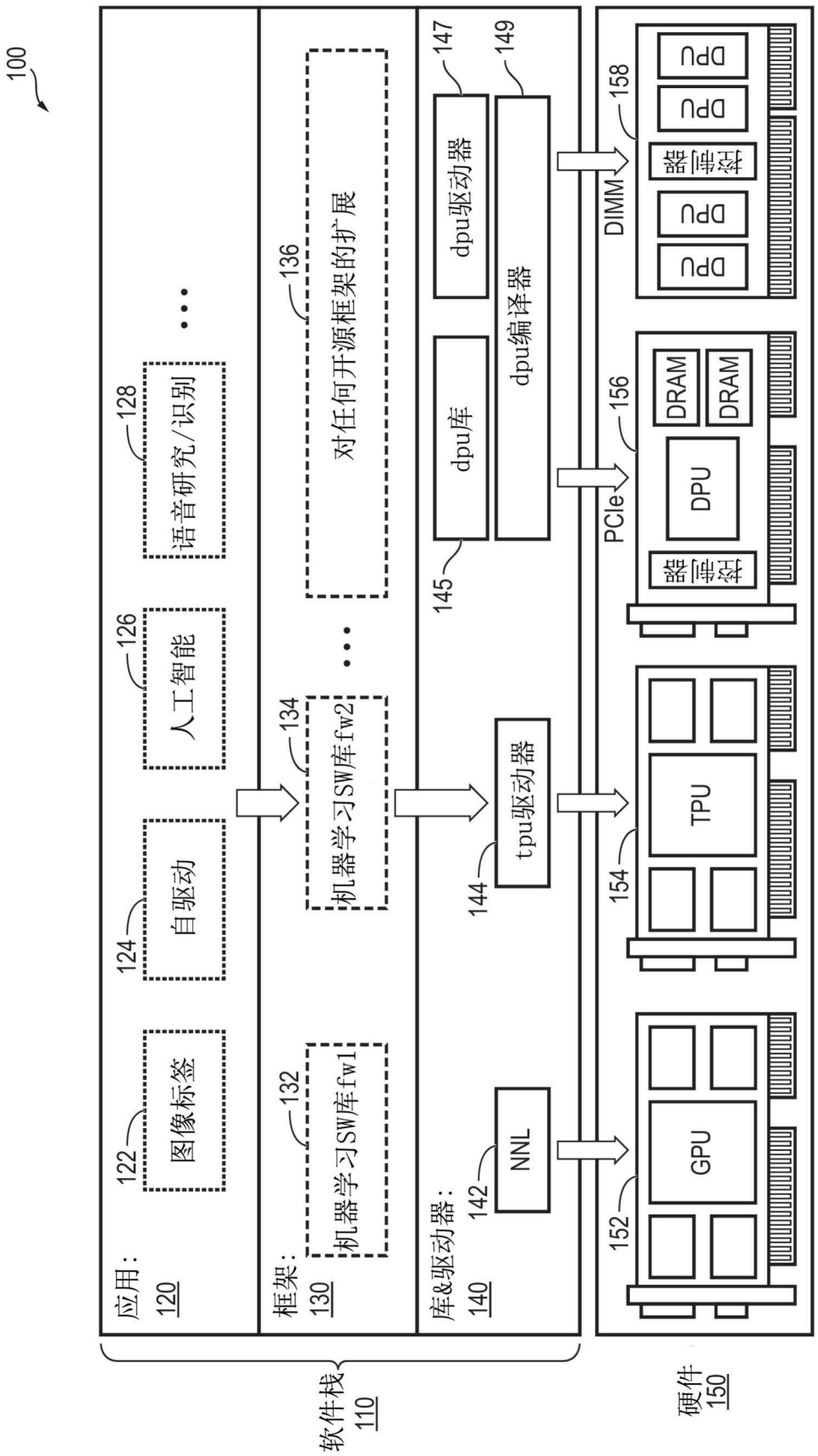

100
应用：
120
图像标签
122
自驱动
124
人工智能
126
语音研究/识别
128
...
框架：
130
机器学习SW库fw1
132
机器学习SW库fw2
134
对任何开源框架的扩展
136
...
库&驱动器：
140
NNL
142
tpu驱动器
144
dpu库
145
dpu编译器
147
dpu驱动器
149
软件栈
110
硬件
150
GPU
152
TPU
154
PCIe
156
DPU
DRAM
DRAM
DIMM
158
DPU
DPU
DPU

图1

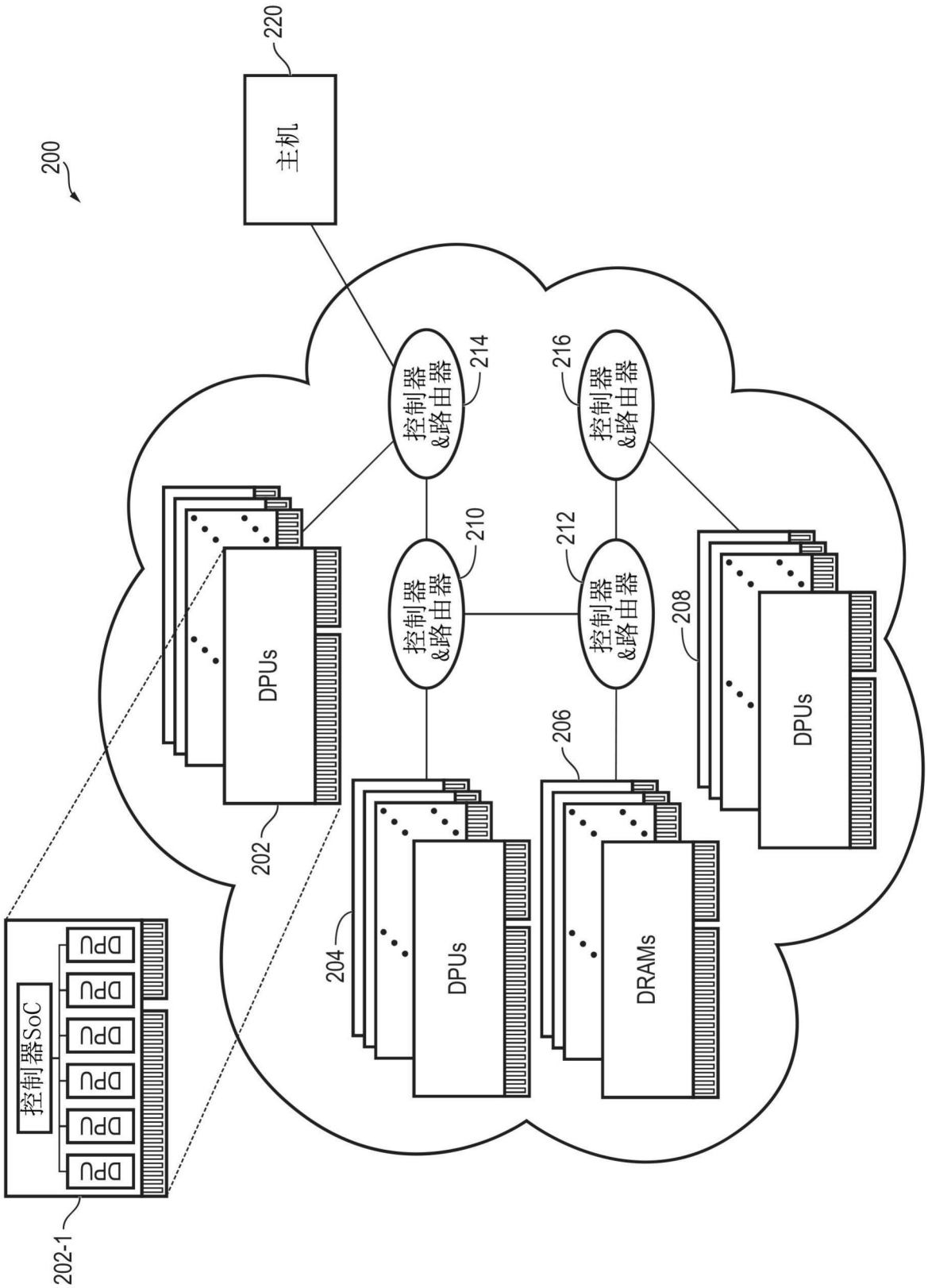

200
220
主机
控制器&路由器
214
216
210
212
208
DPUs
206
204
DRAMs
202
控制器SoC
DPU
202-1

图2

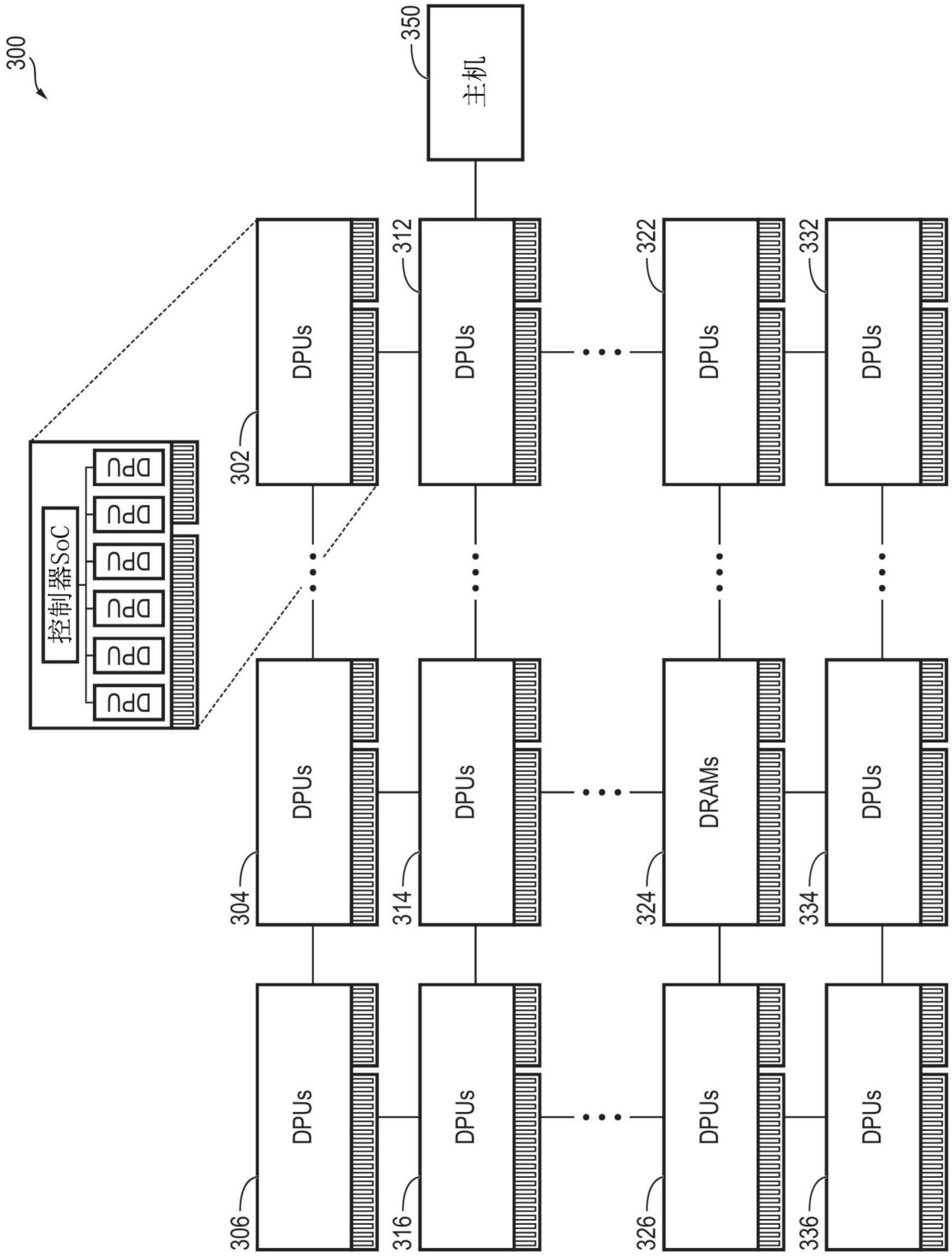

300
350
主机
312
322
332
302
DPUs
控制器SoC
DPU
304
314
324
334
DRAMs
306
316
326
336

图3

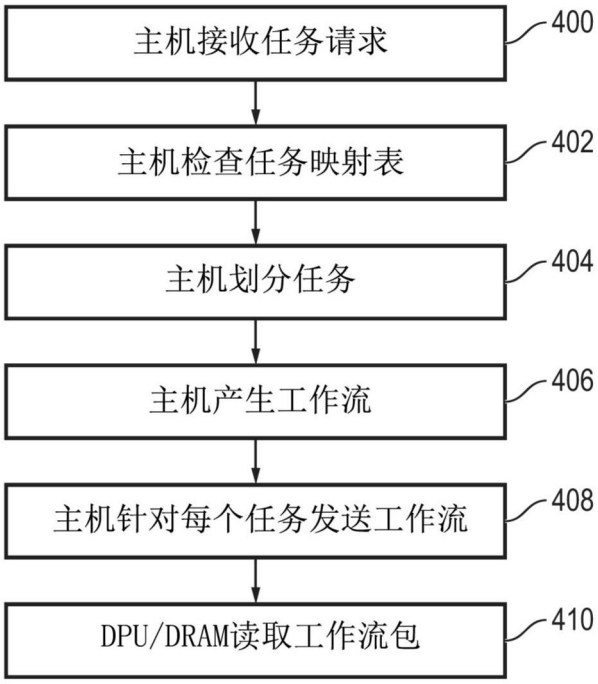

主机接收任务请求
400
主机检查任务映射表
402
主机划分任务
404
主机产生工作流
406
主机针对每个任务发送工作流
408
DPU/DRAM读取工作流包
410

图4

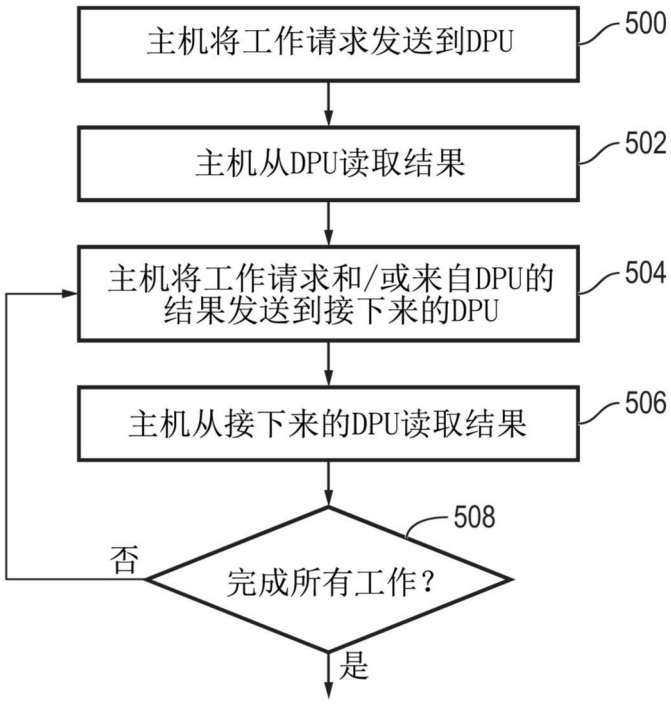

主机将工作请求发送到DPU
500
主机从DPU读取结果
502
主机将工作请求和/或来自DPU的结果发送到接下来的DPU
504
主机从接下来的DPU读取结果
506
508
否
完成所有工作？
是

图5

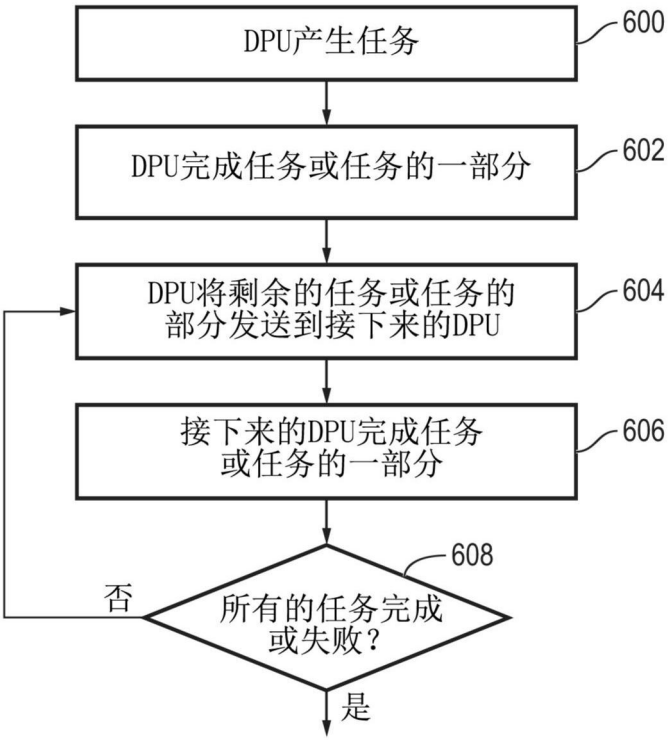

DPU产生任务
600
DPU完成任务或任务的一部分
602
DPU将剩余的任务或任务的部分发送到接下来的DPU
604
接下来的DPU完成任务或任务的一部分
606
608
所有的任务完成或失败？
否
是

图6