



(12) 发明专利申请

(10) 申请公布号 CN 112801125 A

(43) 申请公布日 2021.05.14

(21) 申请号 202011270628.5

G06N 5/04 (2006.01)

(22) 申请日 2020.11.13

(30) 优先权数据

16/682,967 2019.11.13 US

(71) 申请人 辉达公司

地址 美国加利福尼亚州

(72) 发明人 A·马尔雅 J·考茨 李志忠

P·莫尔恰诺夫 H·D·殷

(74) 专利代理机构 北京市磐华律师事务所

11336

代理人 高伟

(51) Int.Cl.

G06K 9/62 (2006.01)

G06N 3/04 (2006.01)

G06N 3/08 (2006.01)

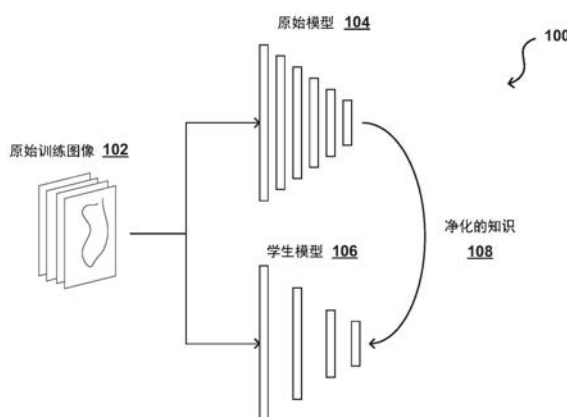
权利要求书3页 说明书86页 附图52页

(54) 发明名称

合成用于训练一个或更多个神经网络的数据

(57) 摘要

公开了合成用于训练一个或更多个神经网络的数据。提出了用于生成对进一步训练神经网络有用的数据的设备、系统和技术。在至少一个实施例中,可以至少部分地基于由一个或更多个神经网络生成的数据,包括先前用于训练所述一个或更多个神经网络的数据,来重新训练所述一个或更多个神经网络。



1. 一种处理器,包括:

一个或多个电路,用于至少部分地基于由一个或多个神经网络生成的数据来帮助重新训练所述一个或多个神经网络,所述数据包括先前用于训练所述一个或多个神经网络的数据。

2. 根据权利要求1所述的处理器,其中由所述一个或多个神经网络生成的所述数据包括在通过所述一个或多个神经网络的多个传递期间修改的一组噪声图像。

3. 根据权利要求2所述的处理器,其中所选分类的激活被最大化以用于通过所述一个或多个神经网络的正向传递,以产生推理和交叉熵损失。

4. 根据权利要求3所述的处理器,其中通过所述一个或多个神经网络的反向传播利用特征分布正则化来维持平均值和方差值。

5. 根据权利要求2所述的处理器,其中根据针对通过所述一个或多个神经网络的所述多个传递确定的噪声梯度修改所述一组噪声图像。

6. 根据权利要求1所述的处理器,其中使用至少一个图像先验进一步修改所述一组噪声图像。

7. 一种系统,包括:

一个或多个处理器,用于至少部分地基于由一个或多个神经网络生成的数据来帮助重新训练所述一个或多个神经网络,所述数据包括先前用于训练所述一个或多个神经网络的数据。

8. 根据权利要求7所述的系统,其中由所述一个或多个神经网络生成的所述数据包括在通过所述一个或多个神经网络的多个传递期间修改的一组噪声图像。

9. 根据权利要求8所述的系统,其中所选分类的激活被最大化以用于通过所述一个或多个神经网络的正向传递,以产生推理和交叉熵损失。

10. 根据权利要求9所述的系统,其中通过所述一个或多个神经网络的反向传播利用特征分布正则化来维持平均值和方差值。

11. 根据权利要求8所述的系统,其中根据针对通过所述一个或多个神经网络的所述多个传递确定的噪声梯度修改所述一组噪声图像。

12. 根据权利要求7所述的系统,其中使用至少一个图像先验进一步修改所述一组噪声图像。

13. 一种方法,包括:

至少部分地基于由一个或多个神经网络生成的数据重新训练所述一个或多个神经网络,所述数据包括先前用于训练所述一个或多个神经网络的数据。

14. 根据权利要求13所述的方法,其中由所述一个或多个神经网络生成的所述数据包括在通过所述一个或多个神经网络的多个传递期间修改的一组噪声图像。

15. 根据权利要求14所述的方法,进一步包括:

其中将所选分类的激活最大化以用于通过所述一个或多个神经网络的正向传递,以产生推理和交叉熵损失。

16. 根据权利要求15所述的方法,进一步包括:

执行通过所述一个或多个神经网络的反向传播,所述反向传播利用特征分布正则化来维持平均值和方差值。

17. 根据权利要求14所述的方法, 其中根据针对通过所述一个或更多个神经网络的所述多个传递确定的噪声梯度修改所述一组噪声图像。

18. 根据权利要求13所述的方法, 其中使用至少一个图像先验进一步修改所述一组噪声图像。

19. 一种在其上存储有一组指令的机器可读介质, 所述一组指令如果由一个或更多个处理器执行, 使得所述一个或更多个处理器至少:

至少部分地基于由一个或更多个神经网络生成的数据重新训练所述一个或更多个神经网络, 所述数据包括先前用于训练所述一个或更多个神经网络的数据。

20. 根据权利要求19所述的机器可读介质, 其中由所述一个或更多个神经网络生成的所述数据包括在通过所述一个或更多个神经网络的多个传递期间修改的一组噪声图像。

21. 根据权利要求20所述的机器可读介质, 其中将所选分类的激活最大化以用于通过所述一个或更多个神经网络的正向传递, 以产生推理和交叉熵损失。

22. 根据权利要求21所述的机器可读介质, 其中通过所述一个或更多个神经网络的反向传播利用特征分布正则化来维持平均值和方差值。

23. 根据权利要求20所述的机器可读介质, 其中根据针对通过所述一个或更多个神经网络的所述多个传递确定的噪声梯度修改所述一组噪声图像。

24. 根据权利要求19所述的机器可读介质, 其中使用至少一个图像先验进一步修改所述一组噪声图像。

25. 一种训练系统, 包括:

一个或更多个处理器, 用于至少部分地基于由一个或更多个神经网络生成的数据来帮助重新训练所述一个或更多个神经网络, 所述数据包括先前用于训练所述一个或更多个神经网络的数据; 以及

存储器, 用于存储经重新训练的一个或更多个神经网络的数据网络参数。

26. 根据权利要求25所述的训练系统, 其中由所述一个或更多个神经网络生成的所述数据包括在通过所述一个或更多个神经网络的多个传递期间修改的一组噪声图像。

27. 根据权利要求26所述的训练系统, 其中所选分类的激活被最大化以用于通过所述一个或更多个神经网络的正向传递, 以产生推理和交叉熵损失。

28. 根据权利要求27所述的训练系统, 其中通过所述一个或更多个神经网络的反向传播利用特征分布正则化以维持平均值和方差值。

29. 根据权利要求26所述的训练系统, 其中根据针对通过所述一个或更多个神经网络的所述多个传递确定的噪声梯度修改所述一组噪声图像。

30. 根据权利要求25所述的训练系统, 其中使用至少一个图像先验进一步修改所述一组噪声图像。

31. 一种处理器, 包括:

一个或更多个算术逻辑单元 (ALU), 用于使用由一个或更多个神经网络生成的合成数据训练所述一个或更多个神经网络, 以对图像数据进行分类, 所述合成数据包括先前用于训练所述一个或更多个神经网络的数据。

32. 根据权利要求31所述的处理器, 其中由所述一个或更多个神经网络生成的数据包括在通过所述一个或更多个神经网络的多个传递期间修改的一组噪声图像。

33.根据权利要求32所述的处理器,其中所选分类的激活被最大化以用于通过所述一个或多个神经网络的正向传递,以产生推理和交叉熵损失。

34.根据权利要求33所述的处理器,其中通过所述一个或多个神经网络的反向传播利用特征分布正则化来维持平均值和方差值。

35.根据权利要求32所述的处理器,其中根据针对通过所述一个或多个神经网络的所述多个传递确定的噪声梯度修改所述一组噪声图像。

36.根据权利要求31所述的处理器,其中使用至少一个图像先验进一步修改所述一组噪声图像。

合成用于训练一个或更多个神经网络的数据

技术领域

[0001] 至少一个实施例涉及用于执行和促进人工智能的处理资源。例如,至少一个实施例涉及用于根据本文描述的各种新颖技术来训练神经网络的处理器或计算系统。

背景技术

[0002] 几乎在每个行业中,机器学习正越来越多地用于各种任务。使用的增加伴随着对适应机器学习模型以满足不断变化的需求的日益增长的需求。对网络的任何修改,其至少需要某种程度的重新训练(retraining),将需要访问原始训练数据,以防止丢失该模型先前学习的内容。不幸的是,原始训练数据并不总是可用,例如可能是隐私或数据访问问题的结果。

附图说明

- [0003] 将参考附图描述根据本公开的各种实施例,其中:
- [0004] 图1A和1B示出了根据至少一个实施例的用于训练学生模型的方法;
- [0005] 图2A、2B和2C示出了根据至少一个实施例的用于合成训练图像的方法;
- [0006] 图3A和3B示出了根据至少一个实施例的经合成的训练图像;
- [0007] 图4示出了根据至少一个实施例的用于合成训练图像的过程;
- [0008] 图5示出了根据至少一个实施例的用于合成训练数据并使用该数据训练模型的过程;
- [0009] 图6示出了根据至少一个实施例的环境;
- [0010] 图7示出了根据至少一个实施例的用于训练可以利用的图像合成网络的系统;
- [0011] 图8示出了根据至少一个实施例的可以利用的统计模型的各层;
- [0012] 图9示出了根据至少一个实施例的推理和/或训练逻辑;
- [0013] 图10示出了根据至少一个实施例的推理和/或训练逻辑;
- [0014] 图11示出了根据至少一个实施例的数据中心系统;
- [0015] 图12示出了根据至少一个实施例的计算机系统;
- [0016] 图13示出了根据至少一个实施例的计算机系统;
- [0017] 图14示出了根据至少一个实施例的计算机系统;
- [0018] 图15示出了根据至少一个实施例的计算机系统;
- [0019] 图16示出了根据至少一个实施例的计算机系统;
- [0020] 图17示出了根据至少一个实施例的计算机系统;
- [0021] 图18示出了根据至少一个实施例的计算机系统;
- [0022] 图19示出了根据至少一个实施例的计算机系统;
- [0023] 图20和图21示出了根据至少一个实施例的共享编程模型;
- [0024] 图22示出了根据至少一个实施例的示例集成电路和相关图形处理器;
- [0025] 图23-图24示出了根据至少一个实施例的示例性集成电路和相关联的图形处理

器；

[0026] 图25-图26示出了根据至少一个实施例的附加的示例性图形处理器逻辑；

[0027] 图27示出了根据至少一个实施例的计算机系统；

[0028] 图28示出了根据至少一个实施例的并行处理器；

[0029] 图29示出了根据至少一个实施例的分区单元；

[0030] 图30示出了根据至少一个实施例的处理集群；

[0031] 图31示出了根据至少一个实施例的图形多处理器；

[0032] 图32示出了根据至少一个实施例的多图形处理单元 (GPU) 系统；

[0033] 图33示出了根据至少一个实施例的图形处理器；

[0034] 图34示出了根据至少一个实施例的处理器器的微架构；

[0035] 图35示出了根据至少一个实施例的深度学习应用处理器；

[0036] 图36示出了根据至少一个实施例的示例神经形态处理器；

[0037] 图37和图38示出了根据至少一个实施例的图形处理器的至少部分；

[0038] 图39示出了根据至少一个实施例的图形处理器核心的至少部分；

[0039] 图40-41示出了根据至少一个实施例的图形处理器核心的至少部分；

[0040] 图42示出了根据至少一个实施例的并行处理单元 (“PPU”)；

[0041] 图43示出了根据至少一个实施例的通用处理集群 (“GPC”)；

[0042] 图44示出了根据至少一个实施例的并行处理单元 (“PPU”) 的内存 分区单元；

[0043] 图45示出了根据至少一个实施例的流式多处理器；

[0044] 图46是根据至少一个实施例的用于高级计算管线的示例数据流程图；

[0045] 图47是根据至少一个实施例的用于在高级计算管线中训练、调整、实例化和部署机器学习模型的示例系统的系统图；

[0046] 图48包括根据至少一个实施例的用于处理成像数据的高级计算管线的示例说明；

[0047] 图49A包括根据至少一个实施例的支持超声设备的虚拟仪器的示例 数据流图示；

[0048] 图49B包括根据至少一个实施例的支持CT扫描仪的虚拟仪器的示例 数据流程图示；

[0049] 图50A示出了根据至少一个实施例的用于训练机器学习模型的过程 的数据流图示；和

[0050] 图50B是根据至少一个实施例的利用预训练的注释模型来增强注释 工具的客户-服务器架构的示例说明。

具体实施方式

[0051] 在至少一个实施例中,可以使用如图1A所示的一组原始训练图像102 来训练机器学习模型,在此称为原始模型104。在至少一个实施例中,原始训练图像102可以包括不同分类的对象的示例,并且可以包括指示这些 相应分类的标签或数据。在至少一个实施例中,一旦被训练,该原始模型 104就可以接收包括对象类型的示例的未分类图像作为输入,并且可以利 用来自对原始训练图像102的训练的提炼知识108来推理在此输入图像中 表示的对象类型的分类。

[0052] 在至少一个实施例中,可能期望修改原始模型104的一方面。在至少一个实施例中,这可以包括修剪原始模型104以产生尺寸较小的学生模型106(例如,经修剪的模型)。在至少一个实施例中,可以执行其他修改,诸如以进一步训练原始模型104(例如,教师模型),从而识别输入图像数据中的对象的附加类别。在至少一个实施例中,可以使用原始训练图像102作为输入以及来自原始模型104的净化知识(distilled knowledge)108来生成学生模型106。在至少一个实施例中,模型106可以是无约束的神经网络。然而,在至少一个实施例中,并且如图1B的布置150所示,当尝试训练学生模型106时,用于训练原始模型104的原始训练图像102可能不可用,或者至少不可访问。在至少一个实施例中,因为对不同数据的训练可能会导致模型忘记或甚至不学习在训练时原始模型104使用原始训练图像102所学习的内容,这可证明是不希望。在至少一个实施例中,仅对附加数据进行进一步训练会导致模型对于使用原始训练图像102学习的分类不再准确的问题。

[0053] 在至少一个实施例中,生成与先前由机器学习模型或神经网络学习的分类有关的合成训练数据。在至少一个实施例中,可以在没有外部实体的帮助或来自外部实体的附加数据的情况下生成该数据。在至少一个实施例中,该合成数据可以用作训练数据,以帮助至少保留那些先前学习的分类的模型或网络的性能。在至少一个实施例中,数据合成方法可以通过经训练的网络反向传播,以尝试将激发在该深度神经网络(DNN)中选定神经元的图像可视化。在至少一个实施例中,这涉及查询使用一组输入训练图像训练的网络以生成使特定分类器(例如,从狗的训练图像中学习的“狗”分类器)产生高正值的输入。在至少一个实施例中,这种技术可以使该经训练的网络产生与先前学习的概念、图案、特征或其他这种关系相对应的图像。在至少一个实施例中,可以尝试确保该网络继续正确地对这些合成图像进行分类,例如当可以在新的或附加的训练数据上对该网络进行训练时。

[0054] 在至少一个实施例中,可以在没有任何合作或来自实体的输入情况下执行合成数据的这种生成,该实体提供了最初(或先前)用于训练该神经网络的一组训练图像(或其他训练数据)。在至少一个实施例中,还可以假定该实体或任何其他外部实体或第三方未提供关于该原始训练数据集、与该原始训练数据集相关联或从该原始训练数据集导出的任何数据。在至少一个实施例中,不需要来自任何形式的先前数据集的图像,诸如完整图像、像素子集或合成的净化图像(distilled image)。在至少一个实施例中,仅使用该数据上训练的网络来生成合成的训练数据,其中可以迫使该网络继续正确地对这些合成的训练数据进行分类,以免忘记任何先前学习的概念。在至少一个实施例中,当先前图像数据不可用时,例如在实体释放经训练的神经网络但由于隐私或其他原因而无法释放训练数据的情况下,可以将新类别添加到对象分类器网络。在至少一个实施例中,实体可能对增加该网络的输出空间感兴趣,以使得能够推理出附加分类。在至少一个实施例中,仅对新类别数据的训练将导致灾难性地忘记先前学习的分类器,并且导致旧类别的性能非常差。在至少一个实施例中,可以生成合成的训练数据,该合成的训练数据可以帮助避免这种遗忘并且能够不断增加该网络的输出空间、修改该网络的大小或进行另一种此类调整。

[0055] 在至少一个实施例中,可以生成表示由网络从先前训练数据中学习到的概念的合成数据。在至少一个实施例中,训练数据合成方法可以使用固定的并且不再学习的经训练的网络204(例如,图1中的原始模型104)来生成表示先前训练数据的示例。在至少一个实施例中,应该能够使用足够大且多样的一组合成图像来重新训练该网络,或训练不同的

网络,并使该网络能够利用这些先前的学习和概念而仍然能够准确地识别这些先前分类中的对象。在至少一个实施例中,这种方法有效地向经训练的网络询问其已学习的概念或在训练期间已接收作为输入的训练数据。在至少一个实施例中,如将关于图2的管线200所讨论的,利用网络激活最大化。在至少一个实施例中,对噪声图像202进行采样并将其作为输入提供给要从中提取概念的经训练的网络204。在至少一个实施例中,噪声图像202将是具有确定的分辨率和纵横比的图像,其中颜色空间中的每个像素的值将是该颜色空间的最小值和最大值之间的随机值,例如许多包括任何适当的输入空间,例如值在0到256之间的RGB颜色空间。可以在正向传递(pass) 208中将此噪声图像202传递通过经训练的神经网络204,并确定网络推理,例如由网络204推理出的分类或标签206。在至少一个实施例中,该标签可以对应于如图2所示的分类“狗”。在至少一个实施例中,对于噪声图像202,可能存在多个潜在的推理,因为它们可能对应于模型204所学习的分类。在至少一个实施例中,可能希望生成用于特定分类的合成训练数据,例如“狗”的分类。在至少一个实施例中,用于目标分类“狗”的输出神经元可以使其激活最大化,或者被设置为尽可能高的值。在至少一个实施例中,可以对该损失函数执行梯度上升,并且该过程可以在该噪声中反向传播210。在至少一个实施例中,最大化针对目标分类的激活可以导致噪声图像开始出现,好像它在该噪声图像中具有目标“狗”分类的对象一样。在至少一个实施例中,该过程在达到目标迭代次数或获得分类最大化的条件之后停止。

[0056] 在至少一个实施例中,可以向该经训练的网络提供一组不同的噪声图像252,具有针对每个图像的激活最大化的相同的目标分类。在至少一个实施例中,在这种最大化激活下的反向传播可以使这些噪声图像254更加类似于目标分类的图像。在至少一个实施例中,这些图像254可以表示学习到的概念,但是仍然可能不显现为目标分类的实际图像。为了生成更能代表实际训练图像的训练图像,同时仍保留所学习到的概念,可以执行至少一些附加处理。

[0057] 在至少一个实施例中,这样的附加处理可以包括基于批归一化(batch norm-based)的正则化。在至少一个实施例中,被训练的网络将为批节点保存多个层的统计信息。在至少一个实施例中,这些特殊层可以用于存储通过该网络传递的激活的均值和方差。在至少一个实施例中,在不强制该图像应遵循相同均值和方差值的情况下生成图像将产生看起来像噪声的图像。在至少一个实施例中,这些图像看起来像噪声,但是对于目标分类给出很高的激活。在至少一个实施例中,通过强制该图像不仅最大化激活,而且该图像还遵循该网络的中间层中的正确批节点统计,可以生成看起来更像目标对象而不太像噪声的图像。在至少一个实施例中,这意味着对于这些图像的激活应当对应于对应的批归一化统计的均值和方差值。在至少一个实施例中,以不同的噪声模式生成多个噪声图像,以便生成多样性的合成训练图像。在至少一个实施例中,可以使用指定数量的图像来获得一组不同的输出图像,例如特定分类的50个不同的输出图像。在至少一个实施例中,可以合成另外的图像,直到满足特定的目标或最终标准为止,例如通过相应的损失值足够小。

[0058] 在至少一个实施例中,输入到经训练的网络的噪声图像252将对应于来自截断(clipped)分布的随机高斯噪声。这些噪声图像将传递通过经训练的神经网络,该神经网络如先前所讨论的那样是固定的,并且将推理出可能对应于这些学习到的分类中的任何分类的输出分类。在至少一个实施例中,可能期望生成用于特定分类的训练图像。在至少

一个实施例中,可以由该网络来推理分类,并且计算交叉熵损失。在至少一个实施例中,图像数据和损失值可以通过该网络反向传播回该输入噪声数据。在至少一个实施例中,由于该网络是固定的,因此不会基于该损失来更新网络权重,而是可以使用噪声梯度来更新该噪声图像。在至少一个实施例中,该过程可以重复许多次迭代,或者通过该经训练的网络传递,直到该噪声图像开始看起来像该目标分类中的对象的真实图像为止。在至少一个实施例中,可以迫使网络(或强烈偏置)以输出指定类别的对象的图像,然后优化输入噪声数据。

[0059] 在至少一个实施例中,可以将各种优化应用于噪声图像。在至少一个实施例中,这可以包括使用一组图像先验,例如其中相邻像素之间的差异受到限制的总变化(total variation)、或者其中针对至少一些向前-向后迭代随机转移输入噪声的抖动。在至少一个实施例中,优化可以产生更清晰的图像256,如图2B所示。在至少一个实施例中,这样的优化可以帮助改善噪声图像的质量,但这对于至少某些情况不是最佳的。在至少一个实施例中,如本文所讨论的,用不同的优化器进行迭代反向传播可以生成合成训练图像258,该合成训练图像258包括该网络学习到的概念并且由充分逼真的并且包括可识别的对象的图像表示。在至少一个实施例中,诸如狗的耳朵和尾巴的特征在这些合成的训练图像中开始可见。

[0060] 在至少一个实施例中,这样的过程可以尝试针对输入 x 进行优化,如下式给出的:

$$[0061] \quad x^* = \operatorname{argmin}_x L(x, y) + R(x)$$

$$[0062] \quad L(x, y) = NLL(\phi_{teacher}(x), y)$$

$$[0063] \quad R(x) = \alpha R_{TV} + \beta R_{l_2}$$

[0064] 在至少一个实施例中,尝试在使对应的损失函数最小化的同时优化 x ,该损失函数在至少一个实施例中是交叉熵或二进制交叉熵以及图像先验项 R 。在至少一个实施例中,此 L 的二进制交叉熵损失方程使输入图像 x 看起来包含目标分类的对象(例如狗)的表示。在至少一个实施例中,可以基于诸如该输入图像的总变化和 L_2 范数(或欧几里得范数)之类的因素来加权可以表示诸如方差和抖动之类的先验的图像先验项 R 。在至少一个实施例中,单独使用先验进行优化可以得到图像300,例如图3A所示的图像。

[0065] 在至少一个实施例中,神经网络可以用作图像先验的近似。在至少一个实施例中,可以针对神经网络中针对一批中图像的每个通道或特征计算均值和方差。在至少一个实施例中,在图像迭代期间,作为特征分布正则化的一部分,该均值和方差可以保持与从最初用于训练的自然图像计算出的均值和方差值尽可能接近。在至少一个实施例中,使用特征分布正则化产生如图3B所示的彩色且至少准逼真的图像350。

[0066] 在至少一个实施例中,为每个通道计算均值和方差值。在至少一个实施例中,可以执行批量归一化。在至少一个实施例中,经训练的网络将已经具有均值和方差值,该均值和方差值用于通过该网络的正向传递。在至少一个实施例中,然后将这些值加到损失函数的正则化因子上。在至少一个实施例中,这可以帮助确保,对于每个通道,在一批数据内,均值与方差值一样,基本上等于来自原始训练的一批范数均值。在至少一个实施例中,可以使用基于其他因素来计算每通道均值和方差值。在至少一个实施例中,不仅可以通过管理输出来获得益处,而且还可以确保中间特征是有意义的,这可以通过对这些中间

特征的统计与从相应输入数据中期望的内容相匹配来实现。在至少一个实施例中,均值和方差信息可以通过下式给出:

$$[0067] \quad R(x) = \alpha R_{TV} + \beta R_{l_2} + \theta R_{BN}$$

$$[0068] \quad R_{BN} = \sum_{i=1}^{\#BN} (\|mean(x_i) - BN_i.mean\|_2 + \|var(x_i) - BN_i.var\|_2)$$

[0069] 在至少一个实施例中,通过经训练的网络的每次迭代可以被认为是反向运行该经训练的网络,以便略微调整该输入噪声图像,从而给定的神经元(或分类)产生更高的置信度得分。在至少一个实施例中,该优化方法类似于反向传播,但是由于该网络是固定的,因此将不调整网络权重,而是对该输入进行调整。在至少一个实施例中,可以将梯度下降独立地应用于输入噪声图像的每个像素。在至少一个实施例中,利用维持均值和方差值的正则化以改善该经修改图像的外观,使其看起来更像实际训练图像。

[0070] 在至少一个实施例中,可以使用图4所示的过程400来合成训练图像。在至少一个实施例中,利用随机噪声(诸如具有高斯噪声分布)生成402输入图像。在至少一个实施例中,如果要生成多个图像,则可以生成不同的随机噪声图像作为输入。在至少一个实施例中,可以确定404或选择分类,针对该分类合成训练图像。在至少一个实施例中,可以在训练期间由神经网络学习了一组分类,并且可以针对这些分类中的每一个合成训练图像。在至少一个实施例中,可以在该经训练的网络中最大化406用于分类的激活值。在至少一个实施例中,该输入图像可以由该网络处理确定的迭代次数。在至少一个实施例中,在通过该网络的正向传递上,可以确定408目标损失值(例如,交叉熵损失)。在至少一个实施例中,然后可以使用特征分布正则化执行410通过该经训练的网络的反向传播,以维持均值和方差值。在至少一个实施例中,一个或更多个图像先验也可以被认为是图像优化过程的一部分。在至少一个实施例中,可以在保持网络参数固定的同时使用噪声梯度来调整412该输入噪声图像。在至少一个实施例中,如果确定414当前图像需要更多的迭代,则该过程可以继续至少另外一次的迭代。在至少一个实施例中,一旦针对当前图像不再有更多迭代,则无论是通过满足最大迭代次数还是满足最终标准,都可以提供416优化图像作为合成训练图像,包括由经训练的网络在先前的训练过程中从原始训练数据中学习到的概念。在至少一个实施例中,该合成图像可以用于重新训练该模型或训练另一模型,而无需访问该原始训练数据。

[0071] 在至少一个实施例中,如图5所示,可以利用用于重新训练模型的过程500。在至少一个实施例中,确定502针对其训练了神经网络的一组分类。在至少一个实施例中,这些分类可以对应于可以在不同图像或视频帧中表示的不同类型的对象。在至少一个实施例中,可以选择504该集合中的针对其合成训练数据的分类。在至少一个实施例中,生成506一组输入图像,每个输入图像具有不同的噪声分布。在至少一个实施例中,可以为图像的像素值生成随机高斯噪声。在至少一个实施例中,可以将这些图像作为输入提供508给该经训练的神经网络。在至少一个实施例中,可以针对当前分类的这些输入噪声图像中的每一个执行510通过该神经网络的多轮迭代。在至少一个实施例中,该网络可以具有针对当前分类最大化的激活,并且可以应用特征分布正则化来实施均值和方差连续性。在至少一个实施例中,网络参数将是固定的,并且反向传播可以替代地导致根据当前噪声梯度来

修改各个噪声图像。在至少一个实施例中,反向传播可以利用在通过该网络的相应正向传递上计算的二进制交叉熵损失。如果确定512 存在更多分类,则该过程可以继续以便生成用于其他分类的图像数据。如果已经针对所有分类生成了图像,则可以将该组图像作为合成训练数据来提供514,该合成训练数据包括通过在原始训练数据集上对该神经网络进行训练而学习到的概念。在至少一个实施例中,这组图像可以用于执行516 训练(或进一步训练)该网络或另一个网络,其中对原始训练数据的访问是不可用的。

[0072] 在至少一个实施例中,这种过程也可以用于网络竞争。在至少一个实施例中,从噪声中生成优化图像将仅与在其上训练该父模型的图像子分类的子集一样好。在至少一个实施例中,可以通过引入学生和教师模型竞赛来鼓励该模型产生更多种图像。在至少一个实施例中,一旦这些图像被反转并且训练了学生模型以在这些反转的图像上工作,就可以生成一组新图像,这组图像对该教师来说是好的,但对该学生来说不是。在至少一个实施例中,可以强制模型反转以生成在先前版本迭代的分布空间中不重叠的图像。在至少一个实施例中,将在附加数据和该先前数据上训练该学生模型。在至少一个实施例中,这可以迭代地执行,并且每次模型反转可以生成一组更加多样化的图像,这最终可以帮助覆盖整个分类空间。

[0073] 在至少一个实施例中,考虑到需要生成大量图像,图像多样性可以起到重要作用,以避免重复而冗余的图像生成。在至少一个实施例中,给定在知识转移期间逐渐改进的学生模型,可以在(a)图像生成过程和(b)学生网络之间采用迭代竞争方案。在至少一个实施例中,可以通过鼓励合成图像引起学生与老师之间的分歧来进行这种竞争。在至少一个实施例中,这导致学生无法容易分类的新图像,而在教师模型可以容易分类。在至少一个实施例中,随着学生继续学习,该过程适应于支持落在学生已经捕获的分布之外同时位于教师的知识范围内的新样本。在至少一个实施例中,如图2C的示例282、284、286、288中所示,该过程可以在学习过程中迭代地扩展图像分布的分布范围。在至少一个实施例中,图2C中的情况280 示出了用于改善图像多样性的适应性竞争方案。在至少一个实施例中,给定一组生成的图像,中学生可以学习捕获原始图像分布的一部分。在至少一个实施例中,在生成新图像时,竞争鼓励学生知识范围之外的新样本,从而改善分布范围。

[0074] 在至少一个实施例中,可以基于Jensen-Shannon散度为图像生成引入额外的损失 R_{compete} ,该散度惩罚(penalize)输出分布相似性,如以下方程所给出的:

$$[0075] \quad R_{\text{compete}}(\hat{x}) = KL(p_T(\hat{x}), M) + KL(p_S(\hat{x}), M)$$

$$[0076] \quad M = 0.5 * (p_T(\hat{x}) + p_S(\hat{x}))$$

[0077] 在至少一个实施例中,Jensen Shannon散度的键合值范围允许与先前的正则项进行有效的联合优化,导致以下形式的整体正则化损失:

$$[0078] \quad \mathcal{R}(\hat{x}) = \mathcal{R}_{\text{prior}}(\hat{x}) + \beta \mathcal{R}_{\text{feature}}(\hat{x}) + \gamma \mathcal{R}_{\text{compete}}(\hat{x})$$

[0079] 其中 β 和 γ 是比例因子。在至少一个实施例中,这增加了用于知识净化设置的DI,对于图像生成适应性地考虑了教师和学生。在至少一个实施例中,当该学生变得更加强大并始终与该老师保持一致时,这两个网络之间的分歧会在以后的阶段鼓励额外的知识转移,从而使知识转移达到令人满意的水平。

[0080] 在至少一个实施例中,可以使用这样的求逆过程来确定由神经网络学习的概念,

而与输入或输出数据的形式无关。在至少一个实施例中，可以合成与音频、视频、语音、文本、运动或物理特性相对应的训练数据。在至少一个实施例中，当至少一些原始训练数据不可用或不可访问时，这样的过程可以用于确定由机器学习模型学习的一个或更多个概念。

[0081] 神经网络训练和部署

[0082] 越来越多的行业和应用正在利用机器学习。在至少一个实施例中，在处理器上开发的深度神经网络 (DNN) 已用于各种用例，从无人驾驶汽车到更快的药物开发，从用于安全系统的自动图像分析到视频聊天应用中的智能实时语言翻译。在至少一个实施例中，深度学习是一种对人脑的神经学习过程进行建模、不断学习、不断变得更加智能以及随着时间的推移更快地提供更准确的结果的技术。最初，成人会教孩子正确识别和分类各种形状，最终无需任何指导即可识别形状。类似地，在至少一个实施例中，将需要对将其设计为完成类似任务的深度学习或神经学习系统进行训练，以使其在识别基本对象、被遮挡对象等时变得更聪明、更有效，同时还向那些对象分配环境。

[0083] 在至少一个实施例中，人脑中的神经元查看接收到的各种输入，将重要性级别分配给这些输入中的每一个，并且将输出传递给其他神经元以对其进行操作。人工神经元或感知器是神经网络的最基本模型。在至少一个实施例中，感知器可以接收表示训练感知器以识别和分类的对象的各个特征的一个或多个输入，并且基于该特征在定义中的重要性，为这些特征中的每一个分配一定的权重对象的形状。

[0084] 深度神经网络 (DNN) 模型包括许多连接的感知器 (例如，节点) 的多层，可以用大量的输入数据对其进行训练，从而以高精度快速解决复杂的问题。在一个示例中，DNN 模型的第一层将汽车的输入图像分解为各个部分，并寻找诸如线条和角之类的基本图案。第二层组装线条以寻找更高级别的图案，例如车轮、挡风玻璃和后视镜。下一层识别车辆的类型，最后几层生成输入图像的标签，识别特定汽车品牌的型号。一旦训练了 DNN，就可以部署该 DNN，并将其用于在称为推理的过程中对对象或模式进行识别和分类。推理示例 (DNN 从给定输入中提取有用信息的过程) 包括识别存入 ATM 机的支票上的手写数字，识别照片中朋友的图像，提供电影推荐，识别和分类不同类型的汽车、行人、以及无人驾驶汽车中的道路危险，或近乎实时地翻译人类语音。

[0085] 在训练期间，数据在正向传播阶段流经 DNN，直到产生指示对应于输入的标签的预测为止。如果神经网络没有正确标记输入，则将分析正确标记和预测标记之间的错误，并在反向传播阶段为每个特征调整权重，直到 DNN 正确标记训练数据集中的输入和其他输入。训练复杂的神经网络需要大量的并行计算性能，包括支持的浮点乘法和加法。推理比训练的计 算密集度小，这是一个对延迟敏感的过程，其中将经过训练的神经网络应用于之前从未见过的新输入，以对图像进行分类、翻译语音和推理新信息。

[0086] 神经网络在很大程度上依赖于矩阵数学运算，而复杂的多层网络需要大量的浮点性能和带宽以提高效率和速度。计算平台拥有数千个处理内核，针对矩阵数学运算进行了优化，并提供数十至数百 TFLOPS 的性能，可以提供基于深度神经网络的人工智能和机器学习应用程序所需的性能。

[0087] 图6示出了在至少一个实施例中的可用于训练和利用机器学习的系统 600 的组件。如将要讨论的，可以由计算设备和资源的各种组合或单个计 算系统来提供各种组件，

其可以在单个实体或多个实体的控制下。此外，方面可以由不同实体触发、发起或请求。在至少一个实施例中，可以由与 供应商环境606相关联的供应商来指导对神经网络的训练，而在至少一个 实施例中，可以由能够通过客户端设备602或其他此类资源访问供应商环境的顾客或其他用户请求训练。在至少一个实施例中，可以由供应商、用 户或第三方内容供应商624提供训练数据(或要由经训练的神经网络分析 的数据)。在至少一个实施例中，客户端设备602可以是要代表用户导航的 车辆或物体，例如，其可以提交请求和/或接收有助于设备导航的指令。

[0088] 在至少一个实施例中，能够通过至少一个网络604提交要被接收到供 应商环境606的请求。在至少一个实施例中，客户端设备可以是任何适当 的电子和/或计算设备，其使得用户能够生成并发送此类请求，如可以包括 台式计算机、笔记本计算机、计算机服务器、智能手机、平板电脑、游戏 机(便携式或其他)、计算机处理器、计算逻辑和机顶盒。网络604可以包 括用于发送请求或其他这样的数据的任何适当的网络，例如可以包括因特 网、内联网、以太网、蜂窝网络、局域网(LAN)、对等点之间的直接无线 连接的网络，等等。

[0089] 在至少一个实施例中，可以将请求接收到接口层608，在该示例中，该接口层可以将数据转发给训练和推理管理器610。该管理器可以是包括 用于管理请求并为相应的数据或内容提供服务的硬件和软件的系统或服务。在至少一个实施例中，该管理器可以接收训练神经网络的请求，并且可以 将针对请求的数据提供给训练管理器612。在至少一个实施例中，训练管 理器612可以选择要使用的适当模型或网络，如果请求未指定，则可以使 用相关的训练数据来训练模型。在至少一个实施例中，训练数据可以是 从客户端设备602接收或从第三方供应商624获得的存储到训练数据存储库 614的一批数据。在至少一个实施例中，训练管理器612可以负责训练数 据，例如通过使用此处讨论的基于LARC的方法。网络可以是任何适当的 网络，例如循环神经网络(RNN)或卷积神经网络(CNN)。一旦训练了 网络并成功评估了网络，就可以将训练后的网络存储到例如模型存储库 616，该模型存储库可以为用户、应用程序或服务存储不同的模型或网络。在至少一个实施例中，可以有多个可以基于许多不同因素使用的单个应用 程序或实体的模型。

[0090] 在至少一个实施例中，在随后的时间点，可以从客户端设备602(或 另一个这样的设备)接收对由经训练的神经网络至少部分地确定或影响的 内容(例如，路径确定)或数据的请求。该请求可以包括例如要使用神经 网络处理的输入数据，以获得一个或更多个推理或其他输出值、分类或预 测。在至少一个实施例中，尽管也可以使用不同的系统或服务，但是输入 数据可以被接收到接口层608并且被引导到推理模块618。在至少一个实 施例中，如果还没有本地存储到推理模块618，则推理模块618可以从模 型存储库616获得合适的经训练的网络，例如这里讨论的经训练的深度神 经网络(DNN)。推理模块618可以将数据作为输入提供给经训练的网络， 然后经训练的网络可以生成一个或多个推理作为输出。例如，这可以包 括 输入数据的实例的分类。在至少一个实施例中，然后可以将推理发送到客 户端设备602以用于向用户显示或其他通信至用户。在至少一个实施例中， 还可以将用于用户的上下文数据存储到用户上下文数据存储库622，该用 户上下文数据存储库622可以包括关于用户的数据，该数据作为网络的输 入对于生成推理或用户获取实例后确定返回给用户的数据是有用的。在至 少一个实施例中，可以包括输入或推理数据中的至少一些的相关数据还可 以被存储到本地数据库620以用于处理将来的请求。在至少一个实施例中，用

户可以使用账户或其他信息来访问供应商环境的资源或功能。在至少一个实施例中,如果允许和可用,则还可以收集用户数据并将其用于进一步的训练模型,以便为将来的请求提供更准确的推理。在至少一个实施例中,可以通过用户界面接收对在客户端设备602上执行机器学习应用626的请求,并通过相同的界面显示结果。客户端设备可以包括诸如处理器628和存储器630之类的资源,用于生成请求和处理结果或响应,以及至少一个数据存储元件632,用于存储机器学习应用626的数据。

[0091] 在至少一个实施例中,处理器628(或训练管理器612的处理器或推理模块618)将是中央处理单元(CPU)。但是,如上所述,这种环境中的资源可以利用GPU来处理至少某些类型的请求的数据。具有数千个内核的GPU被设计为处理大量的并行工作负载,因此在深度学习中已变得很流行,用于训练神经网络和生成预测。尽管使用GPU进行离线构建可以更快地训练更大、更复杂的模型,但离线生成预测意味着无法使用请求时输入特征,或者必须为所有特征排列生成预测并将其存储在查找表中以服务于实时请求。如果深度学习框架支持CPU模式,并且模型又小又简单,足以以合理的延迟在CPU上执行前馈,则CPU实例上的服务可以托管模型。在这种情况下,可以在GPU上离线地进行训练,而在CPU上实时地进行推理。如果CPU方法不可行,则服务可以在GPU实例上运行。但是,由于GPU具有与CPU不同的性能和成本特征,因此运行将运行时算法卸载到GPU的服务可能要求其设计与基于CPU的服务不同。

[0092] 图7示出了至少一个实施例中的可用于对数据进行分类或生成推理的示例系统700。在至少一个实施例中,可以在本文讨论的至少一个实施例中,使用有监督训练和无监督训练。在至少一个实施例中,提供一组训练数据702(例如,分类的或标记的数据)作为输入,以用作训练数据。在至少一个实施例中,训练数据可以包括要为其训练神经网络的至少一种类型的对象的实例,以及标识该类型的对象的信息。在至少一个实施例中,训练数据可以包括一组图像,每个图像包括一种类型对象的表示,其中,每个图像还包括标签、元数据、分类或其他识别相应图像中表示的对象类型的信息片或与之相关联。各种其他类型的数据也可以用作训练数据,并且可以包括文本数据、音频数据、视频数据等。在至少一个实施例中,将训练数据702作为训练输入提供给训练管理器704。在至少一个实施例中,训练管理器704可以是包括用于训练神经网络(或其他模型或算法等)的硬件和软件的系统或服务(例如执行训练应用程序的一个或多个计算设备)。在至少一个实施例中,训练管理器704接收指示将用于训练的模型类型的指令或请求。在至少一个实施例中,模型可以是可用于这种目的的任何适当的统计模型、网络或算法,例如可以包括人工神经网络、深度学习算法、学习分类器、贝叶斯网络等。在至少一个实施例中,训练管理器704可以从适当的存储库706中选择初始模型或其他未训练的模型,并利用训练数据702来训练模型,从而生成可以用于对相似类型的数据进行分类,或生成其他此类推理的经训练的模型708(例如,经训练的深度神经网络)。在不使用训练数据的至少一个实施例中,仍然可以根据训练管理器704选择适当的初始模型来对输入数据进行训练。

[0093] 在至少一个实施例中,可以以多种不同的方式来训练模型,这可以部分地取决于所选择的模型的类型。在至少一个实施例中,可以向机器学习算法提供一组训练数据,其中模型是通过训练过程创建的模型伪像。在至少一个实施例中,训练数据的每个实例包含正确答案(例如,分类),其可以被称为目标或目标属性。在至少一个实施例中,学习算法在

训练数据中 找到将输入数据属性映射到目标的模式、要预测的答案、并且输出捕获这 些模式的机器学习模型。在至少一个实施例中,然后可以使用机器学习模 型来获得针对未指定目标的新数据的预测。

[0094] 在至少一个实施例中,训练管理器704可以从包括二进制分类、多类 别分类和回归模型的一组机器学习模型中进行选择。在至少一个实施例中, 要使用的模型的类型可以至少部分取决于要预测的目标的类型。在至少一 个实施例中,用于二进制分类问题的机器学习模型预测二进制结果,例如 两个可能类别之一。在至少一个实施例中,可以使用诸如 逻辑回归之类的 学习算法来训练二进制分类模型。在至少一个实施例中,用于多类别分类 问题的机器学习模型允许针对多个类别生成预测,诸如以预测多于两个结 果之一。多项式逻辑回归对于训练多类别模型可能很有用。回归问题的机 器学习模型可预测数值。线性回归对于训练回归模型很有用。

[0095] 在至少一个实施例中,为了训练根据一个实施例的机器学习模型,训 练管理器必须确定输入训练数据源以及其他信息,例如包含要预测的目标 的数据属性的名称,所需的数据转换指令以及控制学习算法的训练参数。在至少一个实施例中,在训练过程期间,训 练管理器704可以基于在训练 数据源中指定的目标的类型来自动选择适当的学习算法。在至少一个实施 例中,机器学习算法可以接受用于控制训练过程和所得机器学习模型的某 些属性的参数。这些在本文中称为训练参数。在至少一个实施例中,如果 未指定训练参数, 则训练管理器可以利用已知的默认值来很好地适用于大 范围的机器学习任务。可以为其 指定值的训练参数的示例包括最大模型尺 寸、在训练数据上的最大传递次数、混洗 (shuffle)类型、正则化类型、学习率和正则化量。可以指定默认设置,并带有用于调整值 以微调性能的 选项。

[0096] 在至少一个实施例中,最大模型尺寸是在训练模型期间创建的模式 (pattern)的 总尺寸,以字节为单位。在至少一个实施例中,默认情况下 可以创建指定尺寸的模型,例如 100MB的模型。如果训练管理器无法确 定足够的模式来填充模型大小,则可以创建较小的 模型。如果训练管理器 发现的图案超出了指定大小所能容纳的数量,则可以通过修剪对学习 模型 的质量影响最小的模式来强制实施最大截断。选择模型尺寸提供了对可预 测的模 型质量与使用成本之间的权衡的控制。在至少一个实施例中,较小 的模型可以使训练管理 者移除许多模式以适合最大大小限制,从而影响预 测的质量。在至少一个实施例中,较大的模型查询实时预测可能花费更多。在至少一个实施例中,较大的输入数据集不一定导致 较大的模型,因为模 型存储模式而不是输入数据。在至少一个实施例中,如果模式少且简 单, 则所得模型将很小。具有大量原始属性(输入列)或派生特征(数据转换 的输出)的输 入数据可能会在训练过程中找到并存储更多模式。

[0097] 在至少一个实施例中,训练管理器704可以对训练数据进行多次传递 或迭代以尝 试发现模式。在至少一个实施例中,可以存在默认的传递次数, 例如十次,而在至少一个实 施例中,可以设置多达最大数目的传递,例如 高达一百次传递。在至少一个实施例中,可能 没有最大集合,或者可能存 在会触发训练过程结束的收敛准则或其他因素集合。在至少一 个实施例中, 训练管理器704可以在训练期间监视模式的质量(例如用于模型收敛),并 且 当没有发现更多的数据点或模式时可以自动停止训练。在至少一个实施 例中,仅具有少量 观察的数据集可能需要对数据进行更多传递以获得足够 高的模型质量。较大的数据集可

能包含许多相似的数据点,这可以减少对大量传递的需求。选择对数据的更多数据传递的潜在影响是,模型训练可能会花费更长的时间,并且在资源和系统利用率方面会花费更多。

[0098] 在至少一个实施例中,在训练之前或在训练传递之间对训练数据进行混洗(shuffle)。在至少一个实施例中,混洗是随机的或伪随机的混洗,以产生真正的随机排序,尽管可能存在一些约束以确保不存在某些类型的数据的分组,或者如果存在这种分组,则可以对混洗的数据进行重新混洗。在至少一个实施例中,混洗改变了将数据用于训练的顺序或布置,从而使得训练算法不会遇到相似类型的数据或针对太多陆续观察的单一类型的数据的分组。在至少一个实施例中,可以训练模型以预测对象。在至少一个实施例中,可以在上传之前按对象类型对数据进行排序。在至少一个实施例中,算法然后可以按对象类型按字母顺序处理数据,首先仅遇到特定对象类型的数据。在至少一个实施例中,模型将开始学习该类型的对象的模式。在至少一个实施例中,模型然后将仅遇到第二对象类型的数据,并且将尝试调整模型以适合该对象类型,这会降低适合该第一对象类型的模式。对象类型之间的这种突然转换可能会产生无法学习如何准确预测对象类型的模型。在至少一个实施例中,可以在将训练数据集分为训练和评估子集之前在至少一个实施例中执行混洗,使得对于两个阶段都利用相对均匀的数据类型分布。在至少一个实施例中,训练管理器704可以使用例如伪随机混洗技术来自动混洗数据。

[0099] 在至少一个实施例中,当在至少一个实施例中创建机器学习模型时,训练管理器704可以使用户能够指定设置或应用自定义选项。在至少一个实施例中,用户可以指定一个或更多个评估设置,该评估设置指示要保留的一部分输入数据以用于评估机器学习模型的预测质量。在至少一个实施例中,用户可以指定策略,该策略指示哪些属性和属性转换可用于模型训练。在至少一个实施例中,用户还可以指定控制训练过程和所得模型的某些属性的各种训练参数。

[0100] 在至少一个实施例中,一旦训练管理器已经确定模型的训练完成,例如通过使用本文讨论的至少一个结束准则,则可以提供经训练的模型708,以供分类器714用于分类(或者以其他方式生成推理以用于)验证数据712。在至少一个实施例中,这涉及在模型的训练模式与模型的推理模式之间的逻辑转换。然而,在至少一个实施例中,经训练的模型708将首先被传递到评估器710,该评估器710可以包括在至少一个计算资源(例如,至少一个服务器的CPU或GPU)上执行的用于评估经训练的模型的质量(或其他方面)应用程序、过程或服务。在至少一个实施例中,评估模型以确定该模型在预测新数据和未来数据的目标时是否将提供至少最小的可接受或阈值水平性能。如果否,则训练管理器704可以继续训练该模型。在至少一个实施例中,由于未来的数据实例通常将具有未知的目标值,因此可能希望在已知目标答案的数据上检查机器学习的精度度量,并使用该评估作为在未来的数据上的预测精度的代理。

[0101] 在至少一个实施例中,使用被提供用于训练的训练数据702的子集来评估模型。可以使用如上所述的混洗和拆分方法来确定该子集。在至少一个实施例中,该评估数据子集将被标有目标,因此可以充当评估的地面真实(ground truth)的来源。使用用于训练的相同数据来评估机器学习模型的预测准确性是没有用的,因为对于记住训练数据的模型可能会产生肯定的评估,而不是对其进行概括。在至少一个实施例中,一旦训练完成,就

使用经训练的模型708来处理评估数据子集,并且评估器710可以通过将地面真实数据与该模型的相应输出(或预测/观察)进行比较来确定该模型的准确性。在至少一个实施例中,至少一个实施例中的评估器710可以提供摘要或性能度量,其指示预测值和真实值的匹配程度。在至少一个实施例中,如果训练的模型不满足至少最小性能标准或其他这样的准确性阈值,则可以指示训练管理器704执行进一步的训练,或者在某些情况下,尝试训练新的或不同的模型。在至少一个实施例中,如果经训练的模型708满足相关标准,则可以提供经训练的模型以供分类器714使用。

[0102] 在至少一个实施例中,当创建和训练机器学习模型时,可能希望在至少一个实施例中指定模型设置或训练参数,其将产生能够进行准确预测的模型。在至少一个实施例中,参数包括要执行的传递(正向和/或反向)的数量、正则化或精细化、模型大小和混洗类型。在至少一个实施例中,选择对评估数据产生最佳预测性能的模型参数设置可能导致模型的过拟合。在至少一个实施例中,当模型具有在训练和评估数据源中出现的记忆化(memorized)模式但是未能概括数据中的模式时,发生过度拟合。当训练数据包括评估中使用的所有数据时,经常会发生过度拟合。在至少一个实施例中,已经过拟合的模型在评估期间可能表现良好,但是可能无法对新数据或以其他方式的验证数据做出准确的预测。在至少一个实施例中,为了避免选择过度拟合的模型作为最佳模型,训练管理器可以保留附加数据以验证模型的性能。例如,训练数据集可分为60%用于训练,40%用于评估或验证,可分为两个或更多阶段。在至少一个实施例中,在选择了对于评估数据而言工作很好的模型参数后,导致收敛于诸如该验证数据的一半之类的验证数据的子集之后,可以利用该验证数据的其余部分来执行第二验证以确保这个模型的性能。如果此模型满足对验证数据的期望,则此模型不是过度拟合的数据。在至少一个实施例中,测试集或保持(held-out)集可以用于测试参数。在至少一个实施例中,使用第二验证或测试步骤有助于选择适当的模型参数以防止过度拟合。但是,训练过程中保持更多数据进行验证会使得可用于训练的数据更少。对于较小的数据集,这可能会出现,因为可能没有足够的数据可用于训练。在至少一个实施例中,在这种情况下的一种方法是执行如本文其他地方所讨论的交叉验证。

[0103] 在至少一个实施例中,有许多度量或见解可用于查看和评估给定模型的预测准确性。在至少一个实施例中,评估结果包含用于报告模型的总体成功的预测精度度量,以及有助于探索模型的精度超出预测精度度量的可视化。结果还可以提供查看设置得分阈值(如用于二进制分类)的影响的能力,并可以生成有关标准的警报以检查评估的有效性。度量和可视化的选择可以至少部分取决于要评估的模型的类型。

[0104] 在至少一个实施例中,一旦被令人满意地训练和评估,则可以使用经训练的机器学习模型来构建或支持机器学习应用程序。在一个实施例中,构建机器学习应用程序是涉及一系列步骤的迭代过程。在至少一个实施例中,可以根据观察到的内容和模型要预测的答案来构建核心机器学习问题。在至少一个实施例中,然后可以收集、清理和准备数据以使数据适合于由机器学习模型训练算法使用。可以对这些数据进行可视化和分析,以运行完整性检查,以验证数据质量和理解数据。原始数据(例如输入变量)和答案数据(例如目标)可能没有以可用于训练高度预测模型的方式表示。因此,可能希望从原始变量构建更具预测性的输入表示或特征。可以将结果特征输入到学习算法中,以构建模型并对从模型

构建中保留的数据评估 模型的质量。然后可以使用模型为新数据实例生成目标答案的预测。

[0105] 在至少一个实施例中,在图7的系统700中,在评估之后,经训练的 模型710被提供给分类器714或使其可用,该分类器714能够使用经训练 的模型来处理验证数据。在至少一个实施例中,这可以包括例如从用户或 未分类的第三方接收的数据,诸如寻找关于那些图像中所代表的信息的查 询图像。在至少一个实施例中,可以由分类器使用经训练的模型来处理验 证数据,并且可以将所产生的结果716(例如分类或预测)发送回相应的 源或以其他方式进行处理或存储。在至少一个实施例中,并且在允许这种 使用的情况下,可以将这些现在分类的数据实例存储到训练数据存储库, 训练管理器可以将其用于进一步训练经训练的模型708。在至少一个实施 例中,当可获得新数据时模型将被继续训练,但是在至少一个实施例中, 将根据诸如模型的复杂性或数据集的尺寸等因素定期对这些模型进行重新 训练,例如每天或每周一次。

[0106] 在至少一个实施例中,分类器714可以包括适当的硬件和软件,用于 使用经训练的模型来处理验证数据712。在至少一个实施例中,分类器将 包括一个或更多个计算机服务器,每个计算机服务器具有一个或更多个能 够处理数据的图形处理单元(GPU)。在至少一个实施例中,与CPU或其 他这样的组件相比,GPU的配置和设计可以使它们更期望用于处理 机器学习数据。在至少一个实施例中,可以将至少一个实施例中的训练模型加载 到GPU存储器中,并且将接收到的数据实例提供给GPU以进行处理。GPU 可以具有比CPU多得多的核心,并且GPU核心还可以简单得多。在至少 一个实施例中,给定的GPU可能能够经由不同的硬件线程同时处理数千个 数据实例。在至少一个实施例中,GPU还可以被配置为最大化 浮点吞吐量,这可以为大型数据集提供显著的附加处理优势。

[0107] 在至少一个实施例中,即使当使用GPU、加速器和其他此类硬件来加 速诸如模型训练或使用此类模型进行数据分类之类的任务时,此类任务仍 可能需要大量时间、资源分配和成本。在至少一个实施例中,如果要使用 700个传递来训练机器学习模型,并且数据集包括要用于训练的1,000,000 个数据实例,则将需要为每个传递处理所有百万个实例。架构的不同部分 也可以由不同类型的设备支持。在至少一个实施例中,可以在逻辑上集中的 位置处使用一组服务器来执行训练,如可以作为服务来提供,而原始数 据的分类可以由 诸如服务或在客户端设备上以及其他这样的选项执行。这 些设备也可以由同一实体或多个 实体拥有、操作或控制。

[0108] 在至少一个实施例中,可以训练或以其他方式利用图8中示出的示例 性神经网络 800。在至少一个实施例中,统计模型是人工神经网络(ANN), 其包括多层节点,包括输入层 802、输出层806和中间节点的多个层804, 通常称为“隐藏”层,因为内部层和节点在神经网络中通常不可见或不可 访问。在至少一个实施例中,尽管出于解释的目的仅示出了几个中 间层,但是应当理解的是,对可以利用的中间层的数量没有限制,并且对层的任 何限制通常将是使用模型处理所需的资源或时间的因素。在至少一个实施 例中,除了其他这样的选项之外,还可以使用其他类型的模型、网络、算 法或过程,其可以包括节点和层的其他数量 或选择。在至少一个实施例中,验证数据可以由网络的各层处理以生成一组推理或推理分 数,然后可以将其馈送到损失函数808。

[0109] 在至少一个实施例中,给定层的所有节点互连到相邻层的所有节点。在至少一个

实施例中,中间层的节点然后将分别连接到两个相邻层的节点。在至少一个实施例中,在一些模型中,节点也被称为神经元或连接的单元,并且节点之间的连接被称为边缘。每个节点例如通过使用指定的函数都可以为接收到的输入执行一函数。在至少一个实施例中,节点和边缘可以在训练期间获得不同的权重,并且节点的各个层可以对接收到的输入执行特定类型的变换,其中那些变换也可以在训练期间被学习或调整。在至少一个实施例中,学习可以是有监督的学习或无监督的学习,这可以至少部分取决于训练数据集中包含的信息的类型。在至少一个实施例中,可以利用各种类型的神经网络,例如可以包括卷积神经网络(CNN),其包括多个卷积层和一组池化层,并且已经证明对于诸如图像识别的应用是有益的。由于要确定的参数数量相对较少,因此CNN比其他网络也更易于训练。

[0110] 在至少一个实施例中,可以使用各种调整参数来训练这种复杂的机器学习模型。选择参数、拟合模型和评估模型是模型调整过程各部分,通常称为超参数优化。在至少一个实施例中,这种调整可以包括对基础模型或数据进行内省(introspecting)。在训练或生产设置中,强大的工作流程对于避免超参数的过度拟合至关重要,如本文其他地方所述。交叉验证和向训练数据集添加高斯噪声是可以避免对任何一个数据集过度拟合的有用技术。对于超参数优化,可能需要保持训练和验证集固定不变。在至少一个实施例中,可以在某些类别中调整超参数,例如可以包括数据预处理(例如将单词转换为向量)、CNN架构定义(例如,滤波器大小、滤波器数量)、随机梯度下降(SGD)参数(例如学习率)、正则化或精细化(例如丢弃概率)以及其他此类选项。

[0111] 在至少一个实施例中,可以在预处理期间将数据集的实例嵌入到特定尺寸的低维空间中。在至少一个实施例中,该空间的尺寸是要调整的参数。在至少一个实施例中,CNN架构包含许多可调参数。过滤器尺寸的参数可以表示与要分析的实例的尺寸相对应的信息的解释。在计算语言学中,这称为n-gram(n元)尺寸。示例CNN使用三种不同的过滤器大小,它们代表可能的不同n-gram尺寸。每过滤器尺寸的过滤器数量可以对应于过滤器的深度。每个过滤器都尝试学习与示例结构不同的内容,例如文本数据的句子结构。在卷积层中,激活函数可以是修正的线性单位,并且池化类型设置为最大池化。然后可以将结果串联为单维向量,最后一层完全连接到二维输出。这对应于可以对其应用优化函数的二进制分类。一种这样的函数是梯度下降的均方根(RMS)传播方法的实现,其中示例性的超参数可包括学习率、批尺寸、最大梯度范数和时期(epoch)。使用神经网络,正则化可能是非常重要的考虑因素。在至少一个实施例中,输入数据可以是相对稀疏的。在这种情况下,主要的超参数可能是倒数第二层的丢弃(dropout)参数,其代表了在每个训练周期中不会“开除(fire)”的节点比例。示例训练过程可以基于对先前配置的性能的反馈来建议不同的超参数配置。可以使用推荐的配置来训练模型,评估指定验证集并报告性能。此过程可以重复进行,例如以权衡探索(了解有关不同配置的更多信息)和开发(使用先前的知识以获得更好的结果)。

[0112] 由于训练CNN可以并行化并且可以利用支持GPU的计算资源,因此可以针对不同场景尝试多种优化策略。复杂的场景可以调整模型的架构、预处理和随机梯度下降参数。这扩展了模型配置空间。在基本情况下,仅调整预处理和随机梯度下降参数。与基本方案相比,复杂方案中可以有更多的配置参数。可以使用通过模型的优化环的线性或指数数目的步骤、迭代在关节空间中进行调整。这样的调整过程的成本可以比诸如随机搜索和网络

格搜索之类的调整过程的成本低得多,而不会造成任何明显的性能损失。

[0113] 在至少一个实施例中,可以使用反向传播来计算用于确定神经网络的权重的梯度。反向传播是一种微分形式,如上所述,梯度下降优化算法 可以使用它来调整应用于各种节点或神经元的权重。可以使用相关损失函数的梯度来确定权重。反向传播可以利用损失函数相对于统计模型生成的 输出的导数。如上所述,各种节点可以具有定义各个节点的输出的关联的 激活函数。可以适当使用各种激活函数,包括径向基函数(RBF)和sigmoid,各种支持向量机(SVM)可以使用它们来转换数据。节点的中间层的激活 函数在本文中称为内积内核。这些函数可以包括例如恒等函数、阶跃函数、sigmoidal函数、斜坡函数等等。激活函数也可以是线性或非线性的,以及 其他此类选项。

[0114] 在至少一个实施例中,使用训练数据集训练未训练的神经网络。在 至少一个实施例中,训练框架是PyTorch框架,Tensorflow,Boost,Caffe, Microsoft Cognitive Toolkit/CNTK,MXNet,Chainer,Keras,Deeplearning4j 或其他训练框架。在至少一个实施例中,训练框架训练未训练的神经网络,并使它能够使用本文所述的处理资源来训练,以生成经训练的神经网络。在至少一个实施例中,权重可以被随机选择或者通过使用深度信念网络进 行预训练来选择。在至少一个实施例中,可以有监督、部分有监督或无 监督的方式进行训练。

[0115] 在至少一个实施例中,使用有监督学习来训练未训练的神经网络,其中训练数据集包括与输入的期望输出配对的输入,或者其中训练数据集 包括具有已知输出的输入并且神经网络的输出是手动分级。在至少一个实 施例中,以有监督的方式训练未训练的神经网络,以处理来自训练数据集 的输入,并将所得的输出与一组预期或期望的输出进行比较。在至少一个 实施例中,错误然后通过未训练的神经网络传播回去。在至少一个实施例中,训练框架调整控制未训练的神经网络的权重。在至少一个实施例中, 训练框架包括用于监视未训练的神经网络朝着模型(例如训练的神经网络) 收敛的程度的工具,该模型适合于基于已知输入数据(例如新数据)生成 正确的答案(例如结果)。在至少一个实施例中,训练框架重复训练未训练 的神经网络,同时使用诸如随机梯度下降之类的损失函数和调整算法来调 整权重以细化未训练的神经网络的输出。在至少一个实施例中,训练框架 训练未训练的神经网络,直到未训练的神经网络达到期望的精度为止。在 至少一个实施例中,然后可以部署经训练的神经网络,以实现任何数量的 机器学习操作。

[0116] 至少一个实施例中,未训练的神经网络是使用无监督学习来训练的,其中未训练的神经网络尝试使用未标记的数据来训练自己。在至少一个实 施例中,无监督学习训练数据集将包括输入数据,而不具有任何关联的输 出数据或“地面真实”数据。在至少一个实施例中,未经训练的神经网络 可以学习训练数据集内的分组,并且可以确定各个输入如何与 未经训练的 数据集相关。在至少一个实施例中,可以使用无监督训练来生成自组织图,该 自组织图是一种经训练的神经网络,其能够执行对减少新数据的维数有 用的操作。在至少一个实施例中,无监督训练也可以用于执行异常检测, 其允许识别新数据集中偏离新数据集的正常模式的数据点。

[0117] 在至少一个实施例中,可以使用半监督学习,这是一种在训练数据 集中包括标记数据和未标记数据的混合的技术。在至少一个实施例中,训 练框架可以用于执行增量学习,诸如通过转移的学习技术。在至少一个实 施例中,增量学习使经训练的神经网络能够

适应新数据,而不会忘记在初始训练过程中注入网络中的知识。

[0118] 推理和训练逻辑

[0119] 图9示出了用于执行与一个或多个实施例相关联的推理和/或训练操作的推理和/或训练逻辑915。本文结合图9和/或图10提供关于推理和/或训练逻辑915的细节。

[0120] 在至少一个实施例中,推理和/或训练逻辑915可以包括但不限于代码和/或数据存储901,用于存储前向和/或输出权重和/或输入/输出数据、和/或其他参数以配置在一个或多个实施例的中被训练和/或被用于推理的神经网络的神经元或层。在至少一个实施例中,训练逻辑915可以包括或耦合到代码和/或数据存储901以存储图形代码或其他软件以控制定时和/或顺序,其中权重和/或其他参数信息将被加载以配置包括整数和/或浮点单元(统称为算术逻辑单元(ALU))的逻辑。在至少一个实施例中,诸如图形代码的代码基于代码所对应的神经网络的架构将权重或其他参数信息加载到处理器ALU中。在至少一个实施例中,数据存储901存储在使用一个或多个实施例的方面进行推理和/或训练期间输入/输出数据和/或权重参数的正向传播期间与一个或多个实施例结合训练或使用的神经网络的每一层的权重参数和/或输入/输出数据。在至少一个实施例中,代码和/或数据存储901的任何部分可以与其他片上或片外数据存储(包括处理器的L1、L2或L3高速缓存或系统存储器)一起被包括。

[0121] 在至少一个实施例中,代码和/或数据存储901的任何部分可以在一个或多个处理器或其他硬件逻辑设备或电路的内部或外部。在至少一个实施例中,代码和/或数据存储901可以是高速缓存存储器、动态随机可寻址存储器(“DRAM”)、静态随机可寻址存储器(“SRAM”)、非易失性存储器(例如闪存)或其他存储装置。在至少一个实施例中,代码和/或数据存储901是处理器的内部还是外部的选择,例如,包括DRAM、SRAM、闪存还是由其他类型的存储器,取决于片上可用存储、进行训练和/或推理功能的延迟要求、在推理和/或训练神经网络中使用的数据的批量大小或这些因素的一种组合。在至少一个实施例中,推理和/或训练逻辑915可以包括但不限于代码和/或数据存储905,以存储与在一个或多个实施例的方面中被训练和/或用于推理的神经网络或神经网络的层相对应的向后和/或输出权重和/或输入/输出数据。在至少一个实施例中,代码和/或数据存储905存储在使用一个或多个实施例的方面的训练和/或推理期间在向后传播输入/输出数据和/或权重参数期间,与一个或多个实施例一起训练或结合使用的神经网络的每一层的权重参数和/或输入/输出数据。在至少一个实施例中,训练逻辑915可以包括或耦合到代码和/或数据存储905以存储图形代码或其他软件以控制定时和/或顺序,在该定时和/或顺序中加载权重和/或其他参数信息以配置包括整数和/或浮点单元(统称为算术逻辑单元(ALU))的逻辑。

[0122] 在至少一个实施例中,诸如图形代码的代码基于代码所对应的神经网络的架构将权重或其他参数信息加载到处理器ALU中。在至少一个实施例中,代码和/或数据存储905的任何部分可以与其他片上或片外数据存储包括在一起,包括处理器的L1、L2或L3高速缓存或系统存储器。在至少一个实施例中,代码和/或数据存储905的任何部分可以在一个或多个处理器或其他硬件逻辑设备或电路的内部或外部。在至少一个实施例中,代码和/或数据存储905可以是高速缓存存储器、DRAM、SRAM、非易失性存储器(例如,闪存)或其他存储器。在至少一个实施例中,代码和/或数据存储905是处理器的内部还是外部的选

择,例如,是由DRAM、SRAM、闪存存储器还是其他存储类型组成的,取决于片上可用存储、进行训练和/或推理功能的延迟要求、在推理和/或训练神经网络中使用的数据的批量大小或这些因素的某种组合。

[0123] 在至少一个实施例中,代码和/或数据存储901和代码和/或数据存储 905可以是分开的存储结构。在至少一个实施例中,代码和/或数据存储901 和代码和/或数据存储905可以是相同的存储结构。在至少一个实施例中,代码和/或数据存储901和代码和/或数据存储905可以是部分相同的存储结构和部分分离的存储结构。在至少一个实施例中,代码和/或数据存储901 和代码和/或数据存储905的任何部分可以与其他片上或片外数据存储包括在一起,包括处理器的L1、L2或L3高速缓存或系统存储器。

[0124] 在至少一个实施例中,推理和/或训练逻辑1015可以包括但不限于包括整数和/或浮点单元的一个或更多个算术逻辑单元(“ALU”) 1010,以至少部分地基于训练和/或推理代码(例如图形代码)或由其指示执行逻辑和/或算术操作,其结果可以产生存储在激活存储1020中的作为存储在代码和/或数据存储1001和/或代码和/或数据存储1005中输入/输出和/或权重参数数据功能的激活(例如,来自神经网络内的层或神经元的输出值)。在至少一个实施例中,存储在激活存储1020中的激活是根据由ALU 1010响应于执行指令或其他代码执行的线性代数和/或基于矩阵的数学生成的,其中将存储在代码和/或数据存储1005和/或代码和/或数据存储1001中的权重值与其他值(例如偏置值、梯度信息、动量值或其他参数或超参数)一起用作操作数,这些值中的任意或者全部可以存储在代码和/或数据存储1005或代码和/或数据存储1001或其他片上或片下存储器。

[0125] 在至少一个实施例中,一个或更多个处理器或其他硬件逻辑设备或电路包括一个或更多个ALU 1010,而在另一实施例中,一个或更多个ALU 1010可以在使用它们的处理器或其他硬件逻辑设备或电路(例如,协处理器)的外部。在至少一个实施例中,ALU 1010可以被包括在处理器的执行单元之内,或者以其他方式被包括在处理器的执行单元可以访问的一套ALU中,该套ALU可以在同一处理器内或者分布在不同类型的不同处理器之间(例如,中央处理器、图形处理单元、固定功能单元等)。在至少一个实施例中,代码和/或数据存储1001,代码和/或数据存储1005和激活存储1020可以在同一处理器或其他硬件逻辑装置或电路上,而在另一实施例中,它们可以在不同的处理器或其他硬件逻辑装置或电路中,或相同和不同处理器或其他硬件逻辑设备或电路的某种组合中。在至少一个实施例中,激活存储1020中的任何部分可以与其他片上或片外数据存储包括在一起,包括处理器的L1、L2或L3高速缓存或系统存储器。此外,推理和/或训练代码可以与处理器或其他硬件逻辑或电路可访问的其他代码一起存储,并可以使用处理器的提取、解码、调度、执行、退出和/或其他逻辑电路来提取和/或处理。在至少一个实施例中,激活存储1020可以是高速缓存存储器、DRAM、SRAM、非易失性存储器(例如,闪存)或其他存储器。

[0126] 在至少一个实施例中,激活存储1020可以完全或部分在一个或更多个处理器或其他逻辑电路之内或之外。在至少一个实施例中,激活存储 1020是处理器的内部还是外部的选择,例如,或者是由DRAM、SRAM、闪存存储器或某种其他存储类型组成的,取决于片上可用存储、进行训练和/或推理功能的延迟要求、在推理和/或训练神经网络中使用的数据的批量大小或这些因素的某种组合。在至少一个实施例中,图9中所示的推理和/或训练逻辑1015可以与专用集成电路(“ASIC”) (例如来自谷歌的Tensorflow®处理单元,来自

Graphcore™的推理处理单元 (IPU) 或来自因 特尔公司的 Nervana® (例如“Lake Crest”) 处理器) 结合使用。在至少一个实施例中,图9所示的推理和/或训练逻辑1015可以与中央处理单元 (“CPU”) 硬件,图形处理单元 (“GPU”) 硬件或其他硬件 (例如现场可 编程门阵列 (“FPGA”)) 结合使用。

[0127] 图10示出了根据至少一个或更多个实施例的各种推理和/或训练逻辑1015。在至少一个实施例中,推理和/或训练逻辑1015可以包括但不限于硬件逻辑,其中计算资源是专用的或以其他方式专用地与对应于神经网络内的一层或更多层神经元的权重值或其他信息结合使用。在至少一个实施例中,图10中所示的推理和/或训练逻辑1015可以与专用集成电路 (ASIC) (例如Google的Tensorflow®处理单元、Graphcore™的推理处理单元 (IPU) 或来自因特尔公司的 Nervana® (例如“Lake Crest”) 处理器) 结合使用。在至少一个实施例中,图10所示的推理和/或训练逻辑1015可以与中央处理单元 (CPU) 硬件、图形处理单元 (GPU) 硬件或其他硬件 (例如现场 可编程门阵列 (FPGA)) 结合使用。在至少一个实施例中,推理和/或训练逻辑1015包括但不限于代码和/或数据存储1001和代码和/或数据存储 1005,其可以用于存储代码 (例如图形代码) 权重值和/或其他信息,包括 偏置值、梯度信息、动量值、和/或其他参数或超参数信息。在图10中所 示的至少一个实施例中,数据存储1001和数据存储1005中的每一个都分 别与专用的计算资源 (例如计算硬件1002和计算硬件1006) 相关联。在 至少一个实施例中,计算硬件1002和计算硬件1006中的每一个包括一个 或更多个ALU,一个或更多个ALU仅对分别存储在数据存储1001和代码 和/或数据存储1005中的信息执行数学功能 (例如线性代数函数),其结果 存储在激活存储1020中。

[0128] 在至少一个实施例中,代码和/或数据存储1001和1005以及相应的 计算硬件1002和1006中的每一个分别对应于神经网络的不同层,从而提 供产生来自数代码和/或数据存储1001和计算硬件1002的一个“存储/计算 对1001/1002”的激活作为对下一个代码和/或数据存储1005和计算硬件 1006的“存储/计算对1005/1006”的输入,以镜像神经网络的概念组织。在至少一个实施例中,每个存储/计算对1001/1002和1005/1006可以对应 于一个以上的神经网络层。在至少一个实施例中,在推理和/或训练逻辑 1015中可以包括在存储 计算对1001/1002和1005/1006之后或与之并行的 附加存储/计算对 (未示出)。

[0129] 数据中心

[0130] 图11示出了示例数据中心1100,其中可以使用至少一个实施例。在 至少一个实施例中,数据中心1100包括数据中心基础设施层1110、框架 层1120、软件层1130和应用层 1140。

[0131] 在至少一个实施例中,如图11所示,数据中心基础设施层1110可以 包括资源协调器1112、分组的计算资源1114和节点计算资源 (“节点C.R.”) 1116(1) - 1116(N),其中“N”代 表任何整数、正整数。在至少一个实 施例中,节点C.R.1116(1) - 1116(N) 可以包括但不限于 任何数量的中 央处理单元 (“CPU”) 或其他处理器 (包括加速器、现场可编程门阵列 (FPGA)、图形处理器等)、内存设备 (例如,动态只读存储器)、存储设 备 (例如固态或磁盘驱 动器)、网络输入/输出 (“NW I/O”) 设备、网络交 换机、虚拟机 (“VM”), 电源模块和冷却模块 等。在至少一个实施例中,节点C.R.1116(1) - 1116(N) 中的一个或更多个节点C.R.可以是 具有上 述计算资源的一个或更多个的服务器。

[0132] 在至少一个实施例中,分组的计算资源1114可以包括容纳在一个或更多个机架(未示出)中的节点C.R.的单独分组,或者容纳在各个地理位置(也未示出)的数据中心中的许多机架的单独分组。分组的计算资源1114内的节点C.R.的单独分组可以包括可以配置成或分配为支持一个或更多个工作负载的分组的计算、网络、存储器或存储资源。在至少一个实施例中,可以将包括CPU或处理器的几个节点C.R.分组在一个或更多个机架内,以提供计算资源来支持一个或更多个工作负载。在至少一个实施例中,一个或更多个机架还可以包括任何数量的电源模块、冷却模块和网络交换机的任意组合。

[0133] 在至少一个实施例中,资源协调器1112可以配置成或以其他防水工 会控制一个或更多个节点C.R.1116(1)-1116(N)和/或分组的计算资源 1114。在至少一个实施例中,资源协调器1112可以包括用于数据中心1100 的软件设计基础设施(“SDI”)管理实体。在至少一个实施例中,资源编 排器可以包括硬件、软件或其某种组合。

[0134] 在至少一个实施例中,如图11所示,框架层1120包括作业调度器 1122、配置管理器1124、资源管理器1126和分布式文件系统1128。在至少一个实施例中,框架层1120可以包括用于支持软件层1130的软件1132 和/或应用程序层1140的一个或更多个应用程序1142的框架。在至少一个 实施例中,软件1132或应用程序1142可分别包括基于Web的服务软件或 应用程序,例如由Amazon Web Services,Google Cloud和Microsoft Azure 提供的服务或应用程序。在至少一个实施例中,框架层1120可以是但不限于一种免费和开放源软件网络应用框架,例如可以利用分布式文件系统 1128用于大型扩展数据处理(例如“大数据”)的Apache Spark™(以下称为“Spark”)。在至少一个实施例中,作业调度器1132可以包括Spark 驱动器,以促进对数据中心1100的各个层所支持的工作负荷的调度。在至少一个实施例中,配置管理器1124可以能够配置不同的层(例如包括Spark 的软件层1130和框架层1120)和用于支持大规模数据处理的分布式文件 系统1128。在至少一个实施例中,资源管理器1126能够管理映射到或分 配用于支持分布式文件系统1128和作业调度器1122的集群或分组计算资 源。在至少一个实施例中,集群或分组计算资源可以包括在数据中心基础 设施层1110处的分组计算资源1114。在至少一个实施例中,资源管理器 1126可以与资源协调器1112协调以管理这些映射或分配的计算资源。

[0135] 在至少一个实施例中,包括在软件层1130中的软件1132可以包括由 节点C.R.1116(1)-1116(N)、分组计算资源1114和/或框架层1120的分 布式文件系统1128的至少一部分使用的软件。一种或更多种类型的软件可 以包括但不限于Internet网页搜索软件、电子邮件病毒扫描软件、数据库 软件和流视频内容软件。

[0136] 在至少一个实施例中,应用层1140中包括的应用程序1142可以包括 由节点C.R.1116(1)-1116(N)的至少一部分、分组的计算资源1714和 /或框架层1120的分布式文件系统1128使用的一种或更多种类型的应 用程序。在至少一个实施例中,一种或更多种类型的程序可以包括但不 限于任何数量的基因组学应用程序,认知计算和机器学习应用程序,包 括训练或推理软件,机器学习框架软件(例如PyTorch、TensorFlow、Caffe等)或其他与一个或更多个实施例结合使用的机器学习应用程序。

[0137] 在至少一个实施例中,配置管理器1124、资源管理器1126和资源协 调器1112中的任何一个可以基于以任何技术上可行的方式获取的任何数 量和类型的数据来实现任何数量和类型的自我修改动作。在至少一个实 施例中,自我修改动作可以减轻数据中心1100的

数据中心操作员做出可能不好的配置决定并且可以避免数据中心的未充分利用和/或执行差的部分。

[0138] 在至少一个实施例中,数据中心1100可以包括工具、服务、软件或其他资源,以根据本文所述的一个或更多个实施例来训练一个或更多个机器学习模型或者使用一个或更多个机器学习模型来预测或推理信息。例如,在至少一个实施例中,可以通过使用上文关于数据中心1100描述的软件和计算资源,根据神经网络架构通过计算权重参数来训练机器学习模型。在至少一个实施例中,通过使用通过本文所述的一种或更多种训练技术计算出的权重参数,可以使用上面与关于数据中心1100所描述的资源,使用对应于一个或更多个神经网络的经训练的机器学习模型来推理或预测信息。

[0139] 在至少一个实施例中,数据中心可以使用CPU、专用集成电路(ASIC)、GPU、FPGA或其他硬件来使用上述资源来执行训练和/或推理。此外,上述的一个或更多个软件和/或硬件资源可以配置成一种服务,以允许用户训练或执行信息推理,例如图像识别、语音识别或其他人工智能服务。

[0140] 推理和/或训练逻辑1015用于执行与一个或更多个实施例相关联的推理和/或训练操作。本文结合图9和/或图10提供关于推理和/或训练逻辑1015的细节。在至少一个实施例中,推理和/或训练逻辑1015可以用于推理或预测软件程序以使用优化的决策树解决问题。在至少一个实施例中,推理和/或训练逻辑1015可以用于选择用于解决特定操作的内核。

[0141] 计算机系统

[0142] 图12是示出根据至少一个实施例示例性计算机系统的框图,该示例性计算机系统可以是具有互连的设备和组件的系统,片上系统(SOC)或它们的某种形成有处理器的组合1200,该处理器可以包括执行单元以执行指令。在至少一个实施例中,根据本公开,例如本文所述的实施例,计算机系统1200可以包括但不限于组件,例如处理器1202,其执行单元包括逻辑以执行用于过程数据的算法。在至少一个实施例中,计算机系统1200可以包括处理器,例如可从加利福尼亚圣塔克拉拉的英特尔公司(Intel Corporation of Santa Clara, California)获得的PENTIUM®处理器家族、Xeon™、Itanium®、XScale™和/或StrongARM™, Intel® Core™或Intel® Nervana™微处理器,尽管也可以使用其他系统(包括具有其他微处理器的PC、工程工作站、机顶盒等)。在至少一个实施例中,计算机系统1200可以执行可从华盛顿州雷蒙德市的微软公司(Microsoft Corporation of Redmond, Wash.)获得的WINDOWS操作系统版本,尽管其他操作系统(例如UNIX和Linux)、嵌入式软件和/或图形用户界面也可以使用。

[0143] 实施例可以用在其他设备中,例如手持设备和嵌入式应用。手持设备的一些示例包括蜂窝电话、互联网协议(Internet Protocol)设备、数码相机、个人数字助理(“PDA”)和手持PC。在至少一个实施例中,嵌入式应用可以包括微控制器、数字信号处理器(“DSP”)、片上系统、网络计算机(“NetPC”)、机顶盒、网络集线器、广域网(“WAN”)交换机,或根据至少一个实施例可以执行一个或更多个指令的任何其他系统。

[0144] 在至少一个实施例中,计算机系统1200可包括但不限于处理器1202,该处理器1202可包括但不限于一个或更多个执行单元1208,以根据本文描述的技术执行机器学习模型训练和/或推理。在至少一个实施例中,计算机系统1200是单处理器台式机或服务器

系统,但是在另一实施例中,计算机系统1200可以是多处理器系统。在至少一个实施例中,处理器1202可以包括但不限于复杂指令集计算机(“CISC”)微处理器、精简指令集计算机(“RISC”)微处理器、超长指令字(“VLIW”)微处理器、实现指令集组合的处理器,或任何其他处理器设备,例如数字信号处理器。在至少一个实施例中,处理器1202可以耦合到处理器总线1210,该处理器总线1210可以在处理器1202与计算机系统1200中的其他组件之间传输数据信号。

[0145] 在至少一个实施例中,处理器1202可以包括但不限于1级(“L1”)内部高速缓存存储器(“cache”)1204。在至少一个实施例中,处理器1202可以具有单个内部高速缓存或多级内部缓存。在至少一个实施例中,高速缓存存储器可以驻留在处理器1202的外部。根据特定的实现和需求,其他实施例也可以包括内部和外部高速缓存的组合。在至少一个实施例中,寄存器文件1206可以在各种寄存器中存储不同类型的数据,包括但不限于整数寄存器、浮点寄存器、状态寄存器和指令指针寄存器。

[0146] 在至少一个实施例中,包括但不限于执行整数和浮点运算的逻辑的执行单元1208,其也位于处理器1202中。在至少一个实施例中,处理器1202还可以包括微码(“ucode”)只读存储器(“ROM”),用于存储某些宏指令的微代码。在至少一个实施例中,执行单元1208可以包括用于处理封装指令集1209的逻辑。在至少一个实施例中,通过将封装指令集1209包括在通用处理器1202的指令集中,以及要执行指令的相关电路,可以使用通用处理器1202中的封装数据来执行许多多媒体应用程序使用的操作。在一个或多个实施例中,可以通过使用处理器的数据总线的全宽度来在封装的数据上执行操作来加速和更有效地执行许多多媒体应用程序,这可能不需要在处理器的数据总线上传输较小的数据单元来一次执行一个数据元素的一个或多个操作。

[0147] 在至少一个实施例中,执行单元1208也可以用在微控制器、嵌入式处理器、图形设备、DSP和其他类型的逻辑电路中。在至少一个实施例中,计算机系统1200可以包括但不限于存储器1220。在至少一个实施例中,存储器1220可以被实现为动态随机存取存储器(“DRAM”)设备、静态随机存取存储器(“SRAM”)设备、闪存设备或其他存储设备。在至少一个实施例中,存储器1220可以存储由处理器1202可以执行的由数据信号表示的指令1219和/或数据1221。

[0148] 在至少一个实施例中,系统逻辑芯片可以耦合到处理器总线1210和存储器1220。在至少一个实施例中,系统逻辑芯片可以包括但不限于存储器控制器集线器(“MCH”)1216,并且处理器1202可以经由处理器总线1210与MCH 1216通信。在至少一个实施例中,MCH 1216可以提供到存储器1220的高带宽存储器路径1218以用于指令和数据存储以及用于图形命令、数据和纹理的存储。在至少一个实施例中,MCH 1216可以在处理器1202、存储器1220和计算机系统1200中的其他组件之间启动数据信号,并且在处理器总线1210、存储器1220和系统I/O 1222之间桥接数据信号。

[0149] 在至少一个实施例中,系统逻辑芯片可以提供用于耦合到图形控制器的图形端口。在至少一个实施例中,MCH 1216可以通过高带宽存储器路径1218耦合到存储器1220,并且图形/视频卡1212可以通过加速图形端口(Accelerated Graphics Port)(“AGP”)互连1214耦合到MCH 1216。

[0150] 在至少一个实施例中,计算机系统1200可以使用系统I/O 1222作为专有集线器

接口总线来将MCH 1216耦合到I/O控制器集线器(“ICH”)1230。在至少一个实施例中,ICH 1230可以通过本地I/O总线提供与某些I/O设备的直接连接。在至少一个实施例中,本地I/O总线可以包括但不限于用于将外围设备连接到存储器1220、芯片组和处理器1202的高速I/O总线。示例可以包括但不限于音频控制器1229、固件集线器(“快闪BIOS”)1228、无线收发器1226、数据存储1224、包含用户输入的传统I/O控制器1223和键盘接口、串行扩展端口1227(例如通用串行总线(USB))和网络控制器1234。数据存储装置1224可以包括硬盘驱动器、软盘驱动器、CD-ROM设备、闪存设备或其他大容量存储设备。

[0151] 在至少一个实施例中,图12示出了包括互连的硬件设备或“芯片”的系统,而在其他实施例中,图12可以示出示例性片上系统(“SoC”)。在至少一个实施例中,图中示出的设备可以与专有互连、标准化互连(例如,PCIe)或其某种组合互连。在至少一个实施例中,计算机系统1200的一个或更多个组件使用计算快速链路(CXL)互连来互连。

[0152] 推理和/或训练逻辑1015用于执行与一个或更多个实施例相关的推理和/或训练操作。本文结合图9和/或图10提供关于推理和/或训练逻辑1015的细节。在至少一个实施例中,推理和/或训练逻辑1015可以用于推理或预测软件程序以使用优化的决策树解决问题。在至少一个实施例中,推理和/或训练逻辑1015可以用于选择用于解决特定操作的内核。

[0153] 图13是示出根据至少一个实施例的用于利用处理器1310的电子设备1300的框图。在至少一个实施例中,电子设备1300可以是,例如但不限于,笔记本、塔式服务器、机架服务器、刀片服务器、膝上型计算机、台式机、平板电脑、移动设备、电话、嵌入式计算机或任何其他合适的电子设备。

[0154] 在至少一个实施例中,电子设备1300可以包括但不限于通信地耦合到任何合适数量或种类的组件、外围设备、模块或设备的处理器1310。在至少一个实施例中,处理器1310使用诸如I2C总线、系统管理总线(“SMBus”)、低引脚数(LPC)总线、串行外围接口(“SPI”)、高清音频(“HDA”)总线、串行高级技术附件(“SATA”)总线、通用串行总线(“USB”) (版本1、2、3等)或通用异步接收器/发送器(“UART”)总线之类的总线或接口耦合。在至少一个实施例中,图13示出了系统,该系统包括互连的硬件设备或“芯片”,而在其他实施例中,图13可以示出示例性片上系统(“SoC”)。在至少一个实施例中,图13中示出的设备与专用互连、标准化互连(例如PCIe)或其某种组合互连。在至少一个实施例中,图13的一个或更多个组件使用计算快速链接(CXL)互连进行互连。

[0155] 在至少一个实施例中,图13可以包括显示器1324、触摸屏1325、触摸板1330、近场通信单元(“NFC”)1345、传感器集线器1340、热传感器1346、Express芯片组(“EC”)1335、可信平台模块(“TPM”)1338、BIOS/固件/闪存(“BIOS,FW Flash”)1322、DSP 1360、驱动器1320(例如固态硬盘) (“SSD”)或硬盘驱动器(“HDD”)、无线局域网单元(“WLAN”)1350、蓝牙单元1352、无线广域网单元(“WWAN”)1356、全球定位系统(GPS)单元1355、例如USB 3.0相机的相机(“USB 3.0相机”)1354和/或例如以LPDDR3标准实施的双倍数据速率(“LPDDR”)存储单元(“LPDDR3”)1315。这些组件可以各自以任何合适的方式实现。

[0156] 在至少一个实施例中,其他组件可以通过本文所讨论的组件通信地耦合到处理器1310。在至少一个实施例中,加速度计1341、环境光传感器(“ALS”)1342、罗盘1343和陀螺仪1344可通信地耦合到传感器集线器1340。在至少一个实施例中,热传感器1339、风扇

1337、键盘1346和 触摸板1330可以通信地耦合到EC1335。在至少一个实施例中,扬声器1363、耳机1364和麦克风(“mic”)1365可以通信地耦合到音频单元(“音频 编解码器和D类放大器”)1362,它们又可以通信地耦合到DSP1360。在 至少一个实施例中,音频单元1364可以包括,例如但不限于,音频编码器 /解码器(“codec”)和D类放大器。在至少一个实施例中,SIM卡(“SIM”) 1357可以通信地耦合到WWAN单元1356。在至少一个实施例中,可以以 下一代形状因素(NGFF)实现诸如WLAN单元1350和蓝牙单元1352以 及WWAN单元1356的组件。

[0157] 推理和/或训练逻辑1015用于执行与一个或更多个实施例相关的推 理和/或训练操作。本文结合图9和/或图10提供关于推理和/或训练逻辑 1015的细节。在至少一个实施例中,推理和/或训练逻辑1015可以用于推 理或预测软件程序以使用优化的决策树解决问题。在至少一个实施例中, 推理和/或训练逻辑1015可以用于选择用于解决特定操作的内核。

[0158] 图14示出了根据至少一个实施例的计算机系统1400。在至少一个实 施例中,计算机系统1400被配置为实现贯穿本公开描述的各种过程和方法。

[0159] 在至少一个实施例中,计算机系统1400包括但不限于至少一个中央 处理单元(“CPU”)1402,该中央处理单元1402连接到使用诸如PCI (“外围组件互连”)、外围组件互连快速(“PCI-快速”)、AGP(“加 速图形端口”)、HyperTransport或任何其他总线或点对点通信协议之类 的任何合适协议实现的通信总线1410。在至少一个实施例中,计算机系统 1400包括但不限于主存储器1404和控制逻辑(例如,实现为硬件、软件 或其组合),并且数据被存储在主存储器1404中,其可以采用随机访问存 储器(“RAM”)形式。在至少一个实施例中,网络接口子系统(“网络 接口”)1422提供到其他计算设备和网络的接口,用于从来自 计算机系统 1400的其他系统接收数据或向其他系统传输数据。

[0160] 在至少一个实施例中,计算机系统1400在至少一个实施例中包括但 不限于输入设备1408、并行处理系统1412和可以使用常规阴极射线管 (“CRT”)实现的显示设备1406、液晶显示器(“LCD”)、发光二极 管(“LED”)显示器、等离子显示器或其他合适的显示技术。在至少一 个实施例中,从诸如键盘、鼠标、触摸板、麦克风等的输入设备1408接收 用户输入。在至少一个实施例中,前述模块中的每一个可以位于单个半导 体平台上以形成处理系统。

[0161] 推理和/或训练逻辑1015用于执行与一个或更多个实施例相关的推 理和/或训练操作。本文结合图9和/或图10提供关于推理和/或训练逻辑 1015的细节。在至少一个实施例中,推理和/或训练逻辑1015可以用于推 理或预测软件程序以使用优化的决策树解决问题。在至少一个实施例中, 推理和/或训练逻辑1015可以用于选择用于解决特定操作的内核。

[0162] 图15示出了根据至少一个实施例的计算机系统1500。在至少一个实 施例中,计算机系统1500包括但不限于计算机1510和USB棒1520。在 至少一个实施例中,计算机1510可以包括但不限于任何数量和类型的处理 器(未示出)和一个存储器(未显示)。在至少一个实施例中,计算机1510 包括但不限于服务器、云实例、膝上型计算机和台式计算机。

[0163] 在至少一个实施例中,USB棒1520包括但不限于处理单元1530、USB接口1540和USB接口逻辑1550。在至少一个实施例中,处理单元 1530可以是任何指令执行系统、装置、或能够执行指令的设备。在至少一 个实施例中,处理核心1530可以包括但不限于任何数量

和类型的处理核心（未示出）。在至少一个实施例中，处理核心1530包括专用集成电路（“ASIC”），该专用集成电路被优化为执行与机器学习相关联的任何数量和类型的操作。例如，在至少一个实施例中，处理核心1530是张量处理单元（“TPC”），其被优化以执行机器学习推理操作。在至少一个实施例中，处理单元1530是视觉处理单元（“VPU”），其被优化以执行机器视觉和机器学习推理操作。

[0164] 在至少一个实施例中，USB接口1540可以是任何类型的USB连接器或USB插座。例如，在至少一个实施例中，USB接口1540是用于数据和电源的USB 3.0Type-C插座。在至少一个实施例中，USB接口1540是USB 3.0Type-A连接器。在至少一个实施例中，USB接口逻辑1550可以包括使处理单元1530能够经由USB连接器1540与设备（例如，计算机1510）接口的任何数量和类型的逻辑。

[0165] 推理和/或训练逻辑1015用于执行与一个或多个实施例相关的推理和/或训练操作。本文结合图9和/或图10提供关于推理和/或训练逻辑1015的细节。在至少一个实施例中，推理和/或训练逻辑1015可以用于推理或预测软件程序以使用优化的决策树解决问题。在至少一个实施例中，推理和/或训练逻辑1015可以用于选择用于解决特定操作的内核。

[0166] 图16示出了示例性架构，其中多个GPU 1610-1613通过高速链路1640(1)-1640(N)（例如，总线、点对点互连等）通信地耦合到多个多核处理器1605-1606。在至少一个实施例中，高速链路1640-1643支持4GB/s、30GB/s、80GB/s或更高的通信吞吐量。在至少一个实施例中，可以使用各种互连协议，包括但不限于PCIe 4.0或5.0和NVLink 2.0。

[0167] 另外，在一个实施例中，GPU 1610-1613中的两个或多个通过高速链路1629-1630互连，可以使用与用于高速链路1640-1643的协议/链路相同或不同的协议/链路来实现。类似地，两个或多个多核处理器1605-1606可以通过高速链路1628连接，该高速链路可以是以20GB/s、30GB/s、120 GB/s或更高速度运行的对称多处理器（SMP）总线。可替代地，以使用相似的协议/链路（例如，通过公共互连结构）来完成图16中所示的各种系统组件之间的所有通信。

[0168] 在一个实施例中，每个多核处理器1605-1606分别经由存储器互连1626-1627通信地耦合至处理器存储器1601-1602，并且每个GPU 1610-1613通过GPU存储器互连1650-1653通信地耦合至GPU存储器1620-1623。在至少一个实施例中，存储器互连1626-1627和1650-1653可以利用相似或不同的存储器访问技术。作为示例而非限制，处理器存储器1601-1602和GPU存储器1620-1623可以是易失性存储器，诸如动态随机存取存储器（DRAM）（包括堆叠的DRAM）、图形DDR SDRAM（GDDR）（例如，GDDR5、GDDR6）或高带宽内存（HBM）和/或可以是非易失性存储器（例如3D XPoint或Nano-Ram）。在一个实施例中，处理器存储器1601-1602的某些部分可以是易失性存储器，而另一部分可以是非易失性存储器（例如，使用两级存储器（2LM）层次结构）。

[0169] 如下文所述，尽管各种处理器1605-1606和GPU 1610-1613可以分别物理地耦合到特定的存储器1601-1602、1620-1623，统一的存储器架构可以实现为相同虚拟系统地址空间（也称为“有效地址”空间）分布在各种物理存储器之间。例如，当M=2和N=4时，处理器存储器1601-1602每个可以包括64GB的系统存储器地址空间，并且GPU存储器1620-1623每个可以包括32GB的系统存储器地址空间（在该示例中导致总共256GB的可寻址存储

器)。

[0170] 图17示出了根据一个示例性实施例的用于多核处理器1707和图形加速模块1746之间的互连的附加细节。图形加速模块1746可以包括集成在通过高速链路1740耦合到处理器1707的线卡上的一个或更多个GPU芯片。在至少一个实施例中,图形加速模块1746可以与处理器1707集成在同一封装或芯片上。

[0171] 在至少一个实施例中,所示的处理器1707包括多个核心1760A-1760D,每个核心具有转换后备缓冲器1761A-1761D和一个或更多个高速缓存1762A-1762D。在至少一个实施例中,核心1760A-1760D可以包括用于执行未示出的指令和处理数据的各种其他组件。缓存1762A-1762D可以包括1级(L1)和2级(L2)高速缓存。另外,一个或更多个共享高速缓存1756可以被包括在高速缓存1762A-1762D中,并且由一组核心1760A-1760D共享。例如,处理器1707的一个实施例包括24个核心,每个核心具有其自己的L1高速缓存、十二个共享的L2高速缓存和十二个共享的L3高速缓存。在该实施例中,两个相邻核心共享一个或更多个L2和L3高速缓存。处理器1707和图形加速模块1746与系统存储器1714连接,系统存储器1714可以包括图16的处理器存储器1601-1602。

[0172] 在至少一个实施例中,经由一致性总线1764上的核心间通信,保持存储在各种高速缓存1762A-1762D、1756和系统存储器1714中的数据和指令的一致性。例如,每个高速缓存可具有与其相关联的高速缓存一致性逻辑/电路,该高速缓存一致性逻辑和电路响应于检测到的对特定高速缓存线的读取或写入,在一致性总线1764上进行通信。在一个实现中,在一致性总线1764上实现了缓存监听协议,以监听缓存访问。

[0173] 在至少一个实施例中,代理电路1725将图形加速模块1746通信地耦合到一致性总线1764,从而允许图形加速模块1746作为核心1760A-1760D的对等方参与高速缓存一致性协议。特别地,接口1735通过高速链路1740(例如,PCIe总线、NVLink等)提供到代理电路1725的连接,并且接口1737将图形加速模块1746连接到链路1740。

[0174] 在一个实现中,加速器集成电路1736代表图形加速模块1746的多个图形处理引擎1731、1732、N,提供缓存管理、存储器访问、上下文管理和中断管理服务。在至少一个实施例中,图形处理引擎1731、1732、N,可以各自包括单独的图形处理单元(GPU)。在至少一个实施例中,图形处理引擎1731、1732、N可选地可包括GPU内的不同类型的图形处理引擎,诸如图形执行单元、媒体处理引擎(例如,视频编码器/解码器)、采样器和双轨引擎。在至少一个实施例中,图形加速模块1746可以是具有多个图形处理引擎1731、1732、N的GPU或图形处理引擎1731、1732、N可以是集成在普通封装、线卡或芯片上的单独的GPU。

[0175] 在一个实施例中,加速器集成电路1736包括存储器管理单元(MMU)1739,用于执行各种存储器管理功能,例如虚拟到物理存储器转换(也称为有效到实际存储器转换)和用于访问系统存储器1714的存储器访问协议。MMU 1739还可以包括转换后备缓冲器(TLB)(未示出),用于将虚拟/有效地址转换为物理/实际地址。在一个实现中,高速缓存1738可以存储命令和数据,以供图形处理引擎1731、1732、N有效访问。在一个实施例中,存储在高速缓存1738和图形存储器1733、1734、M中的数据可能使用获取单元1744与核心高速缓存1762A-1762D、1756和系统存储器1714保持一致。如上所述,这可以经由代表高速缓存1738和存储器1733、1734、M的代理电路1725(例如,将更新发送到与处理器高速缓存1762A-1762D、1756上的高速缓存线的修改/访问有关的高速缓存1738并从高速缓存1738

接收更新)来完成。

[0176] 一组寄存器1745存储上下文数据用于由图形处理引擎1731、1732、N(N)执行的线程,并且上下文管理电路1748管理线程上下文。例如,上下文管理电路1748可以执行保存和还原操作以在上下文切换期间保存和还原各个线程的上下文(例如,其中保存第一线程并且存储第二线程,使得第二线程可以由图形处理引擎执行)。例如,在上下文切换上,上下文管理电路1748可以将当前寄存器值存储到存储器中的指定区域(例如,由上下文指针标识)。然后,当返回上下文时,它可以恢复寄存器值。在至少一个实施例中,中断管理电路1747接收并处理从系统设备接收的中断。

[0177] 在一个实现中,MMU 1739将来自图形处理引擎1731的虚拟/有效地址转换为系统存储器1714中的实际/物理地址。在一个实施例中,加速器集成电路1736支持多个(例如4、8、16)图形加速器模块1746和/或其他加速器设备。图形加速器模块1746可以专用于在处理器2307上执行的单个应用程序,或者可以在多个应用程序之间共享。在至少一个实施例中,提出了一种虚拟化图形执行环境,其中图形处理引擎1731、1732、N的资源与多个应用程序或虚拟机(VM)共享。在至少一个实施例中,可以基于与VM和/或应用程序相关联的处理要求和优先级,将资源细分为“切片”,将其分配给不同的VM和/或应用程序。

[0178] 在至少一个实施例中,加速器集成电路1736充当用于图形加速模块1746的系统的桥梁,并提供地址转换和系统存储器高速缓存服务。另外,加速器集成电路1736可以为主机处理器提供虚拟化设施,以管理图形处理引擎1731、1732、N的虚拟化,中断和存储器管理。

[0179] 因为图形处理引擎1731、1732、N的硬件资源被显式映射到主机处理器1707看到的真实地址空间,所以任何主机处理器都可以使用有效地址值直接寻址这些资源。加速器集成电路1736的一个功能是图形处理引擎1731、1732、N的物理分隔,使得它们在系统中表现为独立的单元。

[0180] 在至少一个实施例中,一个或多个图形存储器1733、1734、M分别耦合到图形处理引擎1731、1732、N中的每一个,并且 $N=M$ 。在至少一个实施例中,图形存储器1733-1734存储由每个图形处理引擎1731、1732、N处理的指令和数据。在至少一个实施例中,图形存储器1733、1734、M可以是易失性存储器,例如DRAM(包括堆叠的DRAM)、GDDR存储器(例如,GDDR5、GDDR6)或HBM,和/或可以是非易失性存储器(例如3D XPoint或Nano-Ram)。

[0181] 在一个实施例中,为了减少高速链路1740上的数据流量,使用偏置技术以确保存储在图形存储器1733、1734、M中的数据将是图形处理引擎1731、1732、N和30最频繁使用和核心2360A-2360D最好不要使用(至少不经常使用)的数据。类似地,在至少一个实施例中,偏置机制尝试将核心(最好不是图形处理引擎1731、1732、N)所需的数据保持在核心和系统存储器1714的高速缓存1762A-1762D、1756中。

[0182] 图18示出了另一示例性实施例,其中加速集成电路1836被集成在处理器1807内。在至少这一实施例中,图形处理引擎1831、1832、N经由接口1837和接口1835通过高速链路1840直接与加速器集成电路1836通信(再次,可以利用任何形式的总线或接口协议)。加速器集成电路1836可以执行与关于图17描述的操作相同的操作,但是鉴于其紧密靠近一致性总线1864和高速缓存1862A-1862D、1856,可能具有更高的吞吐量。在至少一个实施例支持不同的编程模型,包括专用-处理编程模型(无图形加速模块虚拟化)和共享编程模型

(具有虚拟化),其可以包括由加速器集成电路1836控制的编程模型和由图形加速模块1846控制的编程模型。

[0183] 在至少一个实施例中,图形处理引擎1831、1832、N专用于单个操作系统下的单个应用程序或处理。在至少一个实施例中,单个应用程序可以将其他应用程序请求集中到图形处理引擎1831、1832、N),从而在VM /分区内提供虚拟化。

[0184] 在至少一个实施例中,图形处理引擎1831、1832、N可以被多个VM /应用程序分区共享。在至少一个实施例中,共享模型可以使用系统管理程序来虚拟化图形处理引擎1831、1832、N以允许每个操作系统的访问。在至少一个实施例中,对于没有管理程序的单个分区系统,操作系统拥有图形处理引擎1831、1832、N。在至少一个实施例中,操作系统可以虚拟化图形处理引擎1831、1832、N以提供对每个进程或应用程序的访问。

[0185] 在至少一个实施例中,图形加速模块1846或单独的图形处理引擎 1831、1832、N使用进程句柄来选择处理元件。在一个实施例中,处理元件被存储在系统存储器1814中,并且可使用本文所述的有效地址到真实地址转换技术来寻址。在至少一个实施例中,进行句柄可以是在向图形处理引擎1831、1832、N注册其上下文时向主机进程提供的特定于实现的值(即,调用系统软件以将处理元件添加到处理元件链表)。在至少一个实施例中,进程句柄的低16位可以是进程元件在链表中的偏移量。

[0186] 图19示出了示例性加速器集成片1990。如本文中所使用的,“片”包括加速集成电路1736(在图17中示出)的处理资源的指定部分。系统存储器1914内的应用程序有效地址空间1982存储处理元件1983。在一个实施例中,处理元件1983响应于处理器1907上执行的应用程序1980的GPU调用1981来存储。处理元件1983包含对应应用程序1980的处理状态。处理元件1983中包含的工作描述符(WD)1984可以是应用程序请求的单个作业或可能包含指向作业队列的指针。在至少一个实施例中,WD 1984是指向应用程序地址空间1982中的作业请求队列的指针。

[0187] 图形加速模块1946和/或单个图形处理引擎1731、1732、N(在图17中示出)可以由系统中的全部或部分进程共享。在至少一个实施例中,可以包括用于设置进程状态并将WD 1984发送到图形加速模块1946以在虚拟化环境中开始作业的基础设施。

[0188] 在至少一个实施例中,专用进程编程模型是特定于实现的。在该模型中,单个进程拥有图形加速模块1946或单个图形处理引擎1731(在图17中示出)。由于图形加速模块1746由单个进程拥有,因此在分配图形加速模块1946时,管理程序为拥有的分区初始化加速器集成电路1736,以及操作系统为拥有进程初始化加速器集成电路1736。

[0189] 在操作中,加速器集成片1990中的WD获取单元1991获取下一个WD 1984,其中包括要由图形加速模块1946的一个或更多个图形处理引擎完成的工作的指示。来自WD 1984的数据可以存储在寄存器1945中和被MMU 1939、中断管理电路1947和/或上下文管理电路1948使用,如图所 示。例如,MMU 1939的一个实施例包括用于访问OS虚拟地址空间1985内的片段/页面表1986的片段/页面漫游电路。中断管理电路1947可以处理从图形加速接收的中断事件1992。当执行图形操作时,由图形处理引擎 1731、1732、N(在图17中示出)生成的有效地址1993被MMU 1939转换为实际地址。

[0190] 在一个实施例中,为每个图形处理引擎1731、1732、N(在图17中示出)和/或图形加速模块1946复制相同的寄存器组1945,并且可以由系统管理程序或操作系统来初始化。

这些重复的寄存器中的每一个都可以包括在加速器集成片1990中。表1中显示了可由管理程序初始化的示例性寄存器。

[0191] 表1-管理程序初始化的寄存器

[0192] [0193]	1	片控制寄存器
	2	真实地址 (RA) 计划进程区指针
	3	授权掩码覆盖寄存器
	4	中断向量表输入偏移
	5	中断向量表入口限制
	6	状态寄存器
	7	逻辑分区 ID
	8	真实地址 (RA) 管理程序加速器利用率记录指正
	9	存储器描述寄存器

[0194] 表2中示出了由操作系统初始化的示例性寄存器

[0195] 表2-操作系统初始化的寄存器

[0196]	1	进程和线程识别
	2	有效地址 (EA) 上下文保存/还原指针
	3	虚拟地址 (VA) 加速器利用率记录指针
	4	虚拟地址 (VA) 存储器段表指针
	5	授权掩码
	6	工作描述符

[0197] 在一个实施例中,每个WD 1984特定于特定的图形加速模块1946 和/或图形处理引擎1731、1732、N(在图17中示出)。它包含图形处理 引擎1731、1732、N(在图17中示出)进行工作所需的所有信息,或者它可以是一个指向存储器位置的指针,在该位置应用程序已经建立了要完成 的工作的命令队列。

[0198] 图20示出了共享模型的一个示例性实施例的附加细节。该实施例包括管理程序真实地址空间2098,其中存储了处理元件列表2099。可通过管 理程序2096访问管理程序实际地址空间2098,管理程序2096虚拟化用于 操作系统2095的图形加速模块引擎。

[0199] 在至少一个实施例中,共享编程模型允许来自系统中所有分区或分 区的子集的全部进程或进程子集使用图形加速模块2046。存在由多个进程 和分区共享(也就是,时间切片共享和图形定向共享)的图形加速模块2046 的两个编程模型。

[0200] 在此模型中,系统管理程序2096拥有图形加速模块2046,并且其功 能可用于所有操作系统2095。为了使图形加速模块2046支持系统管理程 序2096进行虚拟化,图形加速模块2046可以遵循下述:1)应用程序的作 业请求必须是自主的(即,不需要在作业之间保持状态),否则图形加速 模块2046必须提供上下文保存和还原机制。2)图形加速模块2046保证应 用程序的作业请求在指定的时间内完成,包括任何翻译错误,或者图形加 速模块2046提供了抢占作业处理的能力。3)当在定向共享编程模型中操 作时,必须确保图形加速

模块2046在进程之间的公平性。

[0201] 在至少一个实施例中,需要应用程序2080进行具有图形加速模块 2046类型、工作描述符(WD)、权限掩码寄存器(AMR)值和上下文保存/恢复区域指针(CSRP)的操作系统2095系统调用。在至少一个实施例中,图形加速模块类型描述了用于系统调用的目标加速功能。在至少一个实施例中,图形加速模块类型可以是特定于系统的值。在至少一个实施例中,WD是专门针对图形加速模块2046格式化的,并且可以是以下形式:图形加速模块2046命令、指向用户定义结构的有效地址指针、指向命令队列的有效地址指针、或任何其他数据结构以描述要由图形加速模块2046完成的工作。在至少一个实施例中,AMR值是要用于当前进程的AMR状态。在至少一个实施例中,传递给操作系统的值类似于设置AMR的应用程序。在至少一个实施例中,如果加速集成电路1736(在图17中示出)和图形加速模块2046实现不支持用户授权掩码覆盖寄存器(UAMOR),则操作系统可以在管理程序调用中传递AMR之前将当前UAMOR值应用于AMR值。系统管理程序2096可以在将AMR放入处理元件2083之前,可选地应用当前的授权掩码覆盖寄存器(AMOR)值。在至少一个实施例中,CSRP是寄存器2045之一,其中包含应用程序地址空间2082中某个区域的有效地址,以用于图形加速模块2046保存和恢复上下文状态。如果不需要在作业之间保存状态或抢占作业,则此指针是可选的。在至少一个实施例中,上下文保存/恢复区域可以是固定的系统存储器。

[0202] 在接收到系统调用后,操作系统2095可以验证应用程序2080已注册并被授权使用图形加速模块2046。然后,操作系统2095使用表3中所示的信息调用管理程序2096。

[0203] 表3-OS调用信息调用管理程序

[0204]	1	工作描述符(WD)
	2	授权掩码寄存器(AMR)值(可能被屏蔽)
	3	有效地址(EA)上下文保存/还原区域指针(CSRP)
[0205]	4	进程ID(PID)和可选线程ID(TID)
	5	虚拟地址(VA)加速器利用率记录指针(AURP)
	6	存储段表指针的虚拟地址(SSTP)
	7	逻辑中断服务号码(LISN)

[0206] 在接收到管理程序调用之后,管理程序2096验证操作系统2095已注册并被授权使用图形加速模块2046。管理程序2096然后将处理元件2083放入对应的图形加速模块2046类型的处理元件链接列表中。处理元件可能包含表4中所示的信息。

[0207] 表4-处理元件信息

[0208]	1	工作描述符(WD)
	2	授权掩码寄存器(AMR)值(可能被屏蔽)
	3	有效地址(EA)上下文保存/还原区域指针(CSRP)
	4	进程ID(PID)和可选线程ID(TID)
	5	虚拟地址(VA)加速器利用率记录指针(AURP)
	6	存储段表指针的虚拟地址(SSTP)

7	逻辑中断服务号码 (LISN)
8	中断向量表,从管理程序调用参数派生
9	状态寄存器 (SR) 值
10	逻辑分区ID (LPID)
11	真实地址 (RA) 虚拟机管理程序加速器利用率记录指针
12	存储器描述符寄存器 (SDR)

- [0209] 在至少一个实施例中,管理程序初始化多个加速器集成片2090寄存 器2045。
- [0210] 如图21中所示,在至少一个实施例中,使用统一存储器,该统一存 储器可通过用于访问物理处理器存储器2101-2102和GPU存储器 2120-2123的公共虚拟存储器地址空间来寻址。在该实现中,在GPU 2110-2113上执行的操作利用相同的虚拟/有效存储器地址空间来访问处理 器存储器2101-2102,反之亦然,从而简化了可编程性。在一个实施例中, 虚拟/有效地址空间的第一部分被分配给处理器存储器2101、第二部分被分 配给第二处理器存储器2102、第三部分被分配给GPU存储器2120,等等。在至少一个实施例中,整个虚拟/有效存储器空间(有时称为有效地址空间) 由此分布在处理器存储器2101-2102和GPU存储器 2120-2123中的每一个 上,从而允许任何处理器或GPU访问具有映射到该内存的虚拟地址的任何 物理存储器。
- [0211] 在一个实施例中,一个或更多个MMU 2139A-2139E内的偏置/一致 性管理电路 2194A-2194E确保一个或更多个主机处理器(例如2105)和 GPU 2110-2113的缓存之间的缓存一致性,并实施指示在其中应存储某些 类型的数据的物理存储器的偏置技术。尽管在图 21中示出了偏置/一致性 管理电路2194A-2194E的多个实例,可以在一个或更多个主机处 理器2105 的MMU内和/或在加速器集成电路1736(在图17中示出)内实现偏置/ 一致性电 路。
- [0212] 一个实施例允许GPU附加存储器2120-2123被映射为系统存储器的 一部分,并使用共享虚拟存储器 (SVM) 技术进行访问,但不会遭受与完 整系统高速缓存一致性相关的性能缺陷。在至少一个实施例中,将GPU附 加存储器2120-2123作为系统存储器来访问而没有繁重的高速缓存一致性 开销的能力为GPU卸载提供了有益的操作环境。这种安排允许主机 处理器 2105软件设置操作数并访问计算结果,而不会产生传统I/O DMA数据副 本的开销。此类传统副本涉及驱动程序调用,中断和内存映射的I/O (MMIO) 访问,这些访问相对于简单的内存访问而言效率低下。在至少一个实施例 中,在没有高速缓存一致性开销的情况下访问GPU附加存储器2120-2123 的能力对于卸载的执行的执行时间是关键的。例如,在具有大量流传输写 入存储器流量的情况下,缓存一致性开销会大大降低GPU 2110-2113看到的有效写入带宽。在至少一个实施例中,操作数设置的效率,结果访问的 效率以及GPU计算的效率可以在确定GPU卸载的效率中起作用。
- [0213] 在至少一个实施例中,GPU偏置和主处理器偏置的选择由偏差追踪 器数据结构驱动。例如,可以使用偏置表,该偏置表可以是页面粒度结构 (即,以存储器页面的粒度来控制),每个附接到GPU的存储器页面包括 1或2位。在至少一个实施例中,可以在一个或更多 个GPU附加存储器 2120-2123的被盗存储器范围中实现偏置表,在GPU 2110-2113中具有或不具有偏差高速缓存(例如,以高速缓存频繁/最近使用的偏置表的条目)。可选地,可以在 GPU内维护整个偏置表。

[0214] 在至少一个实施例中,在实际访问GPU存储器之前访问与对附接到 GPU附加存储器2120-2123的每次访问相关联的偏置表条目,从而引起以下操作。首先,来自GPU 2110-2113的本地请求(在GPU偏置中找到其页面)被直接转发到相应的GPU存储器2120-2123。来自GPU的在主机偏置中找到其页面的本地请求被转发到处理器2105(例如,如上所述,通过高速链路)。在一个实施例中,来自处理器2105的在主机处理器偏置中找到所请求的页面请求完成了与普通存储器读取类似的请求。可选地,可以将针对GPU偏置的页面的请求转发给GPU 2110-2113。在至少一个实施例中,如果GPU当前不使用页面,则GPU可以将页面转换为主机处理器偏置。在至少一个实施例中,页面的偏置状态可以通过基于软件的机制,通过硬件辅助的基于软件的机制或者在有限的情况下,可以纯粹通过基于硬件的机制来改变。

[0215] 一种用于改变偏置状态的机制采用API调用(例如OpenCL),该调用反过来又调用GPU的设备驱动程序,该设备驱动程序反过来又将消息(或排队命令描述符)发送给GPU,以指导其进行更改偏置状态,并且对于某些过渡,在主机中执行高速缓存刷新操作。在至少一个实施例中,高速缓存刷新操作用于从主机处理器2105偏置到GPU偏置的转变,但是不用于相反的转变。

[0216] 在一个实施例中,通过暂时使主机处理器2105无法缓存GPU偏置的页面,来保持缓存一致性。为了访问这些页面,处理器2105可以请求GPU 2110进行访问,该访问可以立即或可以不立即授权。因此,为了减少处理器2105与GPU 2110之间的通信,确保GPU偏置的页面是GPU所需的页面而不是主机处理器2105所需要的页面是有益的,反之亦然。

[0217] 推理和/或训练逻辑1015用于执行与一个或多个实施例相关的推理和/或训练操作。本文结合图9和/或图10提供关于推理和/或训练逻辑1015的细节。在至少一个实施例中,推理和/或训练逻辑1015可以用于推理或预测软件程序以使用优化的决策树解决问题。在至少一个实施例中,推理和/或训练逻辑1015可以用于选择用于解决特定操作的内核。

[0218] 图22示出了根据本文所述的各种实施例的示例性集成电路和相关联的图形处理器,其可以使用一个或多个IP核心来制造。除了图示之外,在至少一个实施例中可以包括其他逻辑和电路,包括另外的图形处理器/核心、外围接口控制器或通用处理器核。

[0219] 图22是说明根据至少一个实施例的可使用一个或多个IP核心来制造的芯片集成电路2200上的示例性系统的框图。在至少一个实施例中,集成电路2200包括一个或多个应用处理器2205(例如,CPU)、至少一个图形处理器2210,并且可以另外包括图像处理器2215和/或视频处理器2220,他们中的任何一个可能是模块化IP核心。在至少一个实施例中,集成电路2200包括外围或总线逻辑,其包括USB控制器2225、UART控制器2230、SPI/SDIO控制器2235和I²S/I²C控制器2240。在至少一个实施例中,集成电路2200可以包括耦合到高清多媒体接口(HDMI)控制器2250和移动工业处理器接口(MIPI)显示接口2255中的一个或多个的显示设备2245。在至少一个实施例中,存储可以由闪存子系统提供2260,包括闪存和闪存控制器。在至少一个实施例中,可以经由存储器控制器2265提供存储器接口以用于访问SDRAM或SRAM存储器设备。在至少一个实施例中,一些集成电路还包括嵌入式安全引擎2270。

[0220] 推理和/或训练逻辑1015用于执行与一个或多个实施例相关的推理和/或训练

操作。本文结合图9和/或图10提供关于推理和/或训练逻辑 1015的细节。在至少一个实施例中,推理和/或训练逻辑1015可以用于推理或预测软件程序以使用优化的决策树解决问题。在至少一个实施例中,推理和/或训练逻辑1015可以用于选择用于解决特定操作的内核。

[0221] 图23-图24示出了根据本文所述的各个实施例的示例性集成电路和 相关联的图形处理器,其可以使用一个或多个IP核心来制造。除了图示 的之外,在至少一个实施例中还可以包括其他逻辑和电路,包括附加的图形处理器/核心、外围接口控制器或通用处理器核心。

[0222] 图23-图24是示出根据本文描述的实施例的用于在SoC内使用的示例性图形处理器的框图。图23示出了根据至少一个实施例的芯片集成电路上系统的另外示例性图形处理器2310,其可以使用一个或多个IP核心 来制造。图24示出了根据至少一个实施例的芯片集成电路上系统的示例性 图形处理器2340,其可以使用一个或多个IP核心来制造。在至少一个 实施例中,图23的图形处理器2310是低功耗图形处理器核心。在至少一个实施例中,图24的图形处理器2340是更高性能的图形处理器核心。在至少一个实施例中,每个图形处理器2310、2340可以是图22的图形处理器2210的变体。

[0223] 在至少一个实施例中,图形处理器2310包括顶点处理器2305和一个或多个片段处理器2315A-2315N(例如2315A、2315B、2315C、2315D 至2315N-1和2315N)。在至少一个实施例中,图形处理器2310可以经由 单独的逻辑来执行不同的着色器程序,使得顶点处理器2305被优化以执行 针对顶点着色器程序的操作,而一个或多个片段处理器2315A-2315N执 行针对片段的片段(例如,像素)着色操作或像素着色器程序。在至少一个实施例中,顶点处理器2305执行3D图形管线的顶点处理阶段并生成图 元和顶点数据。在至少一个实施例中,一个或多个片段处理器 2315A-2315N使用由顶点处理器2305生成的图元和顶点数据来生成在显 示设备上显示的帧缓冲区。在至少一个实施例中,一个或多个片段处 理器2315A-2315N被优化以执行如在OpenGL API中所提供的片段着色器程 序,其可以用于执行与在Direct 3D API中所提供的像素着色器程序类似的操作。

[0224] 在至少一个实施例中,图形处理器2310附加地包括一个或多个内存管理单元(MMU) 2320A-2320B、高速缓存2325A-2325B和电路互连 2330A-2330B。在至少一个实施例中,一个或多个MMU 2320A-2320B 提供用于图形处理器2310的虚拟到物理地址的映射,包括用于顶点处理器 2305和/或片段处理器2315A-2315N的,除了存储在一个或多个高速缓 存2325A-2325B中的顶点或图像/纹理数据之外,其还可以引用存储在存储器中的顶点或图像/纹理数据。在至少一个实施例中,一个或多个MMU 2320A-2320B可以与系统内的其他MMU同步,包括与图22的一个或多个应用处理器2205、图像处理器2215和/或视频处理器2220相关联的一个 或多个MMU,使得每个处理器2205-2220可以参与共享或统一虚拟存 储器系统。在至少一个实施例中,一个或多个电路互连2330A-2330B使 图形处理器2310能够经由SoC的内部总线或经由直接连接与SoC内的其 他IP核心相接合。

[0225] 在至少一个实施例中,图形处理器2340包括图23的图形处理器的 一个或多个MMU2320A-2320B,高速缓存2325A-2325B和电路互连 2330A-2330B。在至少一个实施例中,包括一个或多个着色器核心 2355A-2355N(例如,2355A、2355B、2355C、2355D、2355E、2355F、直到2355N-1和2355N)的图形图形处理器1340,其提供了统一的着色器 核心架构,

其中单个核心或类型或核心可以执行所有类型的可编程着色器代码,包括用于实现顶点着色器、片段着色器和/或计算着色器的着色器程序代码。在至少一个实施例中,多个着色器核心可以变化。在至少一个实施例中,图形处理器2340包括核心间任务管理器2345,其充当线程分派器,以将执行线程分派给一个或多个着色器核心2355A-2355N和分块单元2358,以加速基于图块的渲染的分块操作,其中在图像空间中细分了场景的渲染操作,例如,以利用场景内的局部空间相干来优化内部高速缓存的使用。

[0226] 推理和/或训练逻辑1015用于执行与一个或多个实施例相关的推理和/或训练操作。本文结合图9和/或图10提供关于推理和/或训练逻辑1015的细节。在至少一个实施例中,推理和/或训练逻辑1015可以用于推理或预测软件程序以使用优化的决策树解决问题。在至少一个实施例中,推理和/或训练逻辑1015可以用于选择用于解决特定操作的内核。

[0227] 图25-图26示出了根据本文描述的实施例的附加的示例性图形处理器逻辑。在至少一个实施例中,图25示出了可以包括在图22的图形处理器2210内的图形核心2500,在至少一个实施例中,其可以是图24中的统一着色器核心2355A-2355N。图26示出了在至少一个实施例中的适用于在多芯片模块上部署的高度并行的通用图形处理单元2530。

[0228] 在至少一个实施例中,图形核心2500包括共享指令高速缓存2502、纹理单元2518和高速缓存/共享存储器2520,它们是图形核心2500内的执行资源所共有的。在至少一个实施例中,图形核心2500可以包括多个切片2501A-2501N或用于每个核心的分区,图形处理器可以包括图形核心2500的多个实例。切片2501A-2501N可以包括支持逻辑,该支持逻辑包括本地指令高速缓存2504A-2504N、线程调度器2506A-2506N、线程分派器2508A-2508N和一组寄存器2510A-2510N。在至少一个实施例中,切片2501A-2501N可以包括一组附加功能单元(AFU 2512A-2512N)、浮点单元(FPU 2514A-2514N)、整数算术逻辑单元(ALU 2516-2516N)、地址计算单元(ACU 2513A-2513N)、双精度浮点单元(DPFPU 2515A-2515N)和矩阵处理单元(MPU 2517A-2517N)。

[0229] 在至少一个实施例中,FPU 2514A-2514N可以执行单精度(32位)和半精度(16位)浮点运算,而DPFPU 2515A-2515N可以执行双精度(64位)浮点运算。在至少一个实施例中,ALU 2516A-2516N可以以8位、16位和32位精度执行可变精度整数运算,并且可以被配置用于混合精度运算。在至少一个实施例中,MPU 2517A-2517N还可被配置用于混合精度矩阵运算,包括半精度浮点运算和8位整数运算。在至少一个实施例中,MPU 2517A-2517N可以执行各种矩阵运算,以加速机器学习应用框架,包括使得能够支持加速的通用矩阵到矩阵乘法(GEMM)。在至少一个实施例中,AFU 2512A-2512N可以执行浮点单元或整数单元不支持的附加逻辑运算,包括三角运算(例如,sine、cosine等)。

[0230] 推理和/或训练逻辑1015用于执行与一个或多个实施例相关的推理和/或训练操作。本文结合图9和/或图10提供关于推理和/或训练逻辑1015的细节。在至少一个实施例中,推理和/或训练逻辑1015可以用于推理或预测软件程序以使用优化的决策树解决问题。在至少一个实施例中,推理和/或训练逻辑1015可以用于选择用于解决特定操作的内核。

[0231] 图26示出了在至少一个实施例中的通用处理单元(GPGPU)2030,其可以被配置为使得高度并行的计算操作能够由图形处理单元阵列来执行。在至少一个实施例中,GPGPU

2030可以直接链接到GPGPU 2030的其他实例,以创建多GPU集群,以提高深度神经网络的训练速度。在至少一个实施例中,GPGPU 2030包括主机接口2032,以实现与主机处理器的连接。在至少一个实施例中,主机接口2032是PCI Express接口。在至少一个实施例中,主机接口2032可以是厂商专用的通信接口或通信结构。在至少一个实施例中,GPGPU 2030从主机处理器接收命令,并使用全局调度器2034将与那些命令相关联的执行线程分发给一组计算集群2036A-2036H。在至少一个实施例中,计算集群2036A-2036H共享高速缓存存储器2038。在至少一个实施例中,高速缓存存储器2038可以用作计算集群2036A-2036H内的高速缓存存储器的高级高速缓存。

[0232] 在至少一个实施例中,GPGPU 2030包括经由一组存储器控制器 2042A-2042B与计算集群2036A-2036H耦合的存储器2044A-2044B。在至少一个实施例中,存储器2044A-2044B可以包括各种类型的存储器设备,包括动态随机存取存储器(DRAM)或图形随机存取存储器,例如同步图形随机存取存储器(SGRAM),包括图形双倍数据速率(GDDR)存储器。

[0233] 在至少一个实施例中,计算集群2036A-2036H各自包括一组图形核心,诸如图25的图形核心2500,其可以包括多种类型的整数和浮点逻辑单元,其可以包括适用于机器学习计算的精度范围执行计算操作。例如,在至少一个实施例中,每个计算集群2036A-2036H中的浮点单元的至少一个子集可以配置为执行16位或32位浮点运算,而浮点单元的不同子集可以配置为执行64位浮点运算。

[0234] 在至少一个实施例中,GPGPU 2030的多个实例可以被配置为操作为计算集群。在至少一个实施例中,计算集群2036A-2036H用于同步和数据交换的通信在实施例之间变化。在至少一个实施例中,GPGPU 2030的多个实例通过主机接口2032进行通信。在至少一个实施例中,GPGPU 2030包括I/O集线器2039,其将GPGPU 2030与GPU链路2040耦合,其实现了至GPGPU 2030的其他实例的直接连接。在至少一个实施例中,GPU链路2040耦合到专用GPU到GPU桥,其使得能够在GPGPU 2030的多个实例之间进行通信和同步。在至少一个实施例中,GPU链路2040与高速互连耦合,以向其他GPGPU或并行处理器发送和接收数据。在至少一个实施例中,GPGPU 2030的多个实例位于单独的数据处理系统中,并经由可经由主机接口2032访问的网络设备进行通信。在至少一个实施例中,GPU链路2040可被配置为使得能够连接到主机处理器,附加或替代主机接口 2032。

[0235] 在至少一个实施例中,GPGPU 2030可以配置为训练神经网络。在至少一个实施例中,可以在推理平台内使用GPGPU 2030。在其中使用GPGPU 2030进行推理的至少一个实施例中,相对于当使用GPGPU训练神经网络时,GPGPU可以包括更少的计算集群2036A-2036H。在至少一个实施例中,与存储器2044A-2044B相关联的存储器技术可以在推理和训练配置之间有所不同,其中更高带宽的存储器技术专用于训练配置。在至少一个实施例中,GPGPU 2030的推理配置可以支持推理特定指令。例如,在至少一个实施例中,推理配置可以提供对一个或多个8位整数点积指令的支持,该指令可以在部署的神经网络的推理操作期间使用。

[0236] 推理和/或训练逻辑1015用于执行与一个或多个实施例相关的推理和/或训练操作。本文结合图9和/或图10提供关于推理和/或训练逻辑 1015的细节。在至少一个实施例中,推理和/或训练逻辑1015可以用于推理或预测软件程序以使用优化的决策树解决问题。在至少一个实施例中,推理和/或训练逻辑1015可以用于选择用于解决特定操作的内

核。

[0237] 图27是示出根据至少一个实施例的计算系统2700的框图。在至少一个实施例中，计算系统2700包括处理子系统2701，其具有一个或更多个处理器2702和经由互连路径通信的系统存储器2704，互连路径可以包括存储器集线器2705。在至少一个实施例中，存储器集线器2705可以是芯片组组件内的单独组件，也可以集成在一个或更多个处理器2702内。在至少一个实施例中，存储器集线器2705经由通信链路2706与I/O子系统2711耦合。在至少一个实施例中，I/O子系统2711包括I/O集线器2707，其可以使计算系统2700能够接收来自一个或更多个输入设备2708的输入。在至少一个实施例中，I/O集线器2707可以使能显示控制器，其可以包括在一个或更多个处理器2702中，用于向一个或更多个显示设备2710A提供输出。在至少一个实施例中，与I/O集线器2707耦合的一个或更多个显示设备2710A可以包括本地、内部或嵌入式显示设备。

[0238] 在至少一个实施例中，处理子系统2701包括经由总线或其他通信链路2713耦合到存储器集线器2705的一个或更多个并行处理器2712。在至少一个实施例中，通信链路2713可以是许多基于标准的通信链路技术或协议中的一种，例如但不限于PCI快速，或者可以是供应商特定的通信接口或通信结构。在至少一个实施例中，一个或更多个并行处理器2712形成计算集中的并行或向量处理系统，其可以包括大量的处理核心和/或处理集群，例如集成多核心(MIC)处理器。在至少一个实施例中，一个或更多个并行处理器2712形成可以将像素输出到经由I/O集线器2707耦合的一个或更多个显示设备2710A之一的图形处理子系统。在至少一个实施例中，一个或更多个并行处理器2712还可以包括显示控制器和显示接口(未示出)，以使得能够直接连接到一个或更多个显示设备2710B。

[0239] 在至少一个实施例中，系统存储单元2714可以连接到I/O集线器2707，以提供用于计算系统2700的存储机制。在至少一个实施例中，I/O开关2716可以用于提供接口机制，以实现I/O集线器2707与其他组件之间的连接，例如可以集成到平台中的网络适配器2718和/或无线网络适配器2719，以及可以经由一个或更多个附加设备2720添加的各种其他设备。在至少一个实施例中，网络适配器2718可以是以以太网适配器或另一有线网络适配器。在至少一个实施例中，无线网络适配器2719可以包括Wi-Fi、蓝牙、近场通信(NFC)中的一个或更多个或包括一个或更多个无线电的其他网络设备。

[0240] 在至少一个实施例中，计算系统2700可以包括未明确示出的其他组件，包括USB或其他端口连接、光存储驱动器、视频捕获设备等，也可以连接到I/O集线器2707。在至少一个实施例中，对图27中的各个组件进行互连的通信路径可以使用任何合适的协议来实现，诸如基于PCI(外围组件互连)的协议(例如，PCI-快速)，或其他总线或点对点通信接口和/或协议(例如，NV-链路高速互连或互连协议)。

[0241] 在至少一个实施例中，一个或更多个并行处理器2712包括针对图形和视频处理而优化的电路(包括例如视频输出电路)，并构成图形处理单元(GPU)。在至少一个实施例中，一个或更多个并行处理器2712包括针对通用处理而优化的电路。在至少一个实施例中，计算系统2700的组件可以与单个集成电路上的一个或更多个其他系统元件集成。例如，在至少一个实施例中，一个或更多个并行处理器2712、存储器集线器2705、处理器2702和I/O集线器2707可以被集成到片上系统(SoC)集成电路中。在至少一个实施例中，计算系统2700的组件可以被集成到单个封装中以形成系统级封装(SIP)配置。在至少一个实

施例中,计算系统2700的组件的至少一部分可以被集成到多芯片模块(MCM)中,该多芯片模块可以与其他多芯片模块互连到模块化计算系统中。

[0242] 推理和/或训练逻辑1015用于执行与一个或多个实施例相关的推理和/或训练操作。本文结合图9和/或图10提供关于推理和/或训练逻辑1015的细节。在至少一个实施例中,推理和/或训练逻辑1015可以用于推理或预测软件程序以使用优化的决策树解决问题。在至少一个实施例中,推理和/或训练逻辑1015可以用于选择用于解决特定操作的内核。

[0243] 处理器

[0244] 图28示出了根据至少一个实施例的并行处理器2800。在至少一个实施例中,并行处理器2800的各种组件可以使用一个或多个集成电路设备来实现,例如可编程处理器、专用集成电路(ASIC)或现场可编程门阵列(FPGA)。在至少一个实施例中,所示的并行处理器2800是根据示例性实施例的图27所示的一个或多个并行处理器2712的变体。

[0245] 在至少一个实施例中,并行处理器2800包括并行处理单元2802。在至少一个实施例中,并行处理单元2802包括I/O单元2804,其使得能够与其他设备进行通信,包括并行处理单元2802的其他实例。在至少一个实施例中,I/O单元2804可以直接连接到其他设备。在至少一个实施例中,I/O单元2804通过使用集线器或交换机接口(例如,存储器集线器2805)与其他设备连接。在至少一个实施例中,存储器集线器2805与I/O单元2804之间的连接形成通信链路2813。在至少一个实施例中,I/O单元2804与主机接口2806和存储器交叉开关2816连接,其中主机接口2806接收用于执行处理操作的命令,而存储器交叉开关2816接收用于执行存储器操作的命令。

[0246] 在至少一个实施例中,当主机接口2806经由I/O单元2804接收命令缓冲区时,主机接口2806可以引导工作操作以执行那些命令到前端2808。在至少一个实施例中,前端2808与调度器2810耦合,调度器2810配置成将命令或其他工作项分配给处理集群阵列2812的集群。在至少一个实施例中,调度器2810确保在将任务分配给处理集群阵列2812之前,处理集群阵列2812被正确地配置并且处于有效状态。在至少一个实施例中,调度器2810通过在微控制器上执行的固件逻辑来实现。在至少一个实施例中,微控制器实现的调度器2810可配置成以粗粒度和细粒度执行复杂的调度和工作分配操作,从而实现对在处理阵列2812上执行的线程的快速抢占和环境切换。在至少一个实施例中,主机软件可以证明用于通过多个图形处理门铃(doorbell)之一在处理阵列2812上进行调度的工作负载。在至少一个实施例中,工作负载然后可以由包括调度器2810的微控制器内的调度器2810逻辑在处理阵列2812上自动分配。

[0247] 在至少一个实施例中,处理集群阵列2812可以包括多达“N”个处理集群(例如,集群2814A、集群2814B到集群2814N)。在至少一个实施例中,处理集群阵列2812的每个集群2814A-2814N可以执行大量并发线程。在至少一个实施例中,调度器2810可以使用各种调度和/或工作分配算法将工作分配给处理集群阵列2812的集群2814A-2814N,其可以根据每种程序或计算类型产生的工作负载而变化。在至少一个实施例中,调度可以由调度器2810动态地处理,或者可以在配置为由处理集群阵列2812执行的程序逻辑的编译期间部分地由编译器逻辑来辅助。在至少一个实施例中,可将处理集群阵列2812的不同的集群2814A-2814N分配用于处理不同类型的程序或用于执行不同类型的计算。

[0248] 在至少一个实施例中,处理集群阵列2812可以配置成执行各种类型的并行处理操作。在至少一个实施例中,处理集群阵列2812配置成执行通用并行计算操作。例如,在至少一个实施例中,处理集群阵列2812可以包括执行处理任务的逻辑,该处理任务包括对视频和/或音频数据的过滤,执行建模操作,包括物理操作以及执行数据转换。

[0249] 在至少一个实施例中,处理集群阵列2812配置成执行并行图形处理操作。在至少一个实施例中,处理集群阵列2812可以包括附加逻辑以支持这种图形处理操作的执行,包括但不限于执行纹理操作的纹理采样逻辑,以及镶嵌逻辑和其他顶点处理逻辑。在至少一个实施例中,处理集群阵列2812可以配置成执行与图形处理有关的着色器程序,例如但不限于顶点着色器、曲面细分着色器、几何着色器和像素着色器。在至少一个实施例中,并行处理单元2802可以经由I/O单元2804从系统存储器传送数据以进行处理。在至少一个实施例中,在处理期间,可以在处理期间将传送的数据存储到片上存储器(例如,并行处理器存储器2828),然后将其写回到系统存储器。

[0250] 在至少一个实施例中,当并行处理单元2802用于执行图形处理时,调度器2810可以配置成将处理工作负载划分为近似相等大小的任务,以更好地将图形处理操作分配给处理集群阵列2812的多个集群2814A-2814N。在至少一个实施例中,处理集群阵列2812的部分可以配置成执行不同类型的处理。例如,在至少一个实施例中,第一部分可以配置成执行顶点着色和拓扑生成,第二部分可以配置成执行镶嵌和几何着色,并且第三部分可以配置成执行像素着色或其他屏幕空间操作,以生成用于显示的渲染图像。在至少一个实施例中,可以将由集群2814A-2814N中的一个或多个产生的中间数据存储在缓冲区中,以允许在集群2814A-2814N之间传输中间数据以进行进一步处理。

[0251] 在至少一个实施例中,处理集群阵列2812可以经由调度器2810接收要执行的处理任务,该调度器2810从前端2808接收定义处理任务的命令。在至少一个实施例中,处理任务可以包括要被处理的数据的索引,例如表面(补丁)数据、原始数据、顶点数据和/或像素数据,以及状态参数和定义如何处理数据的命令(例如,要执行什么程序)。在至少一个实施例中,调度器2810可以配置成获取与任务相对应的索引,或者可以从前端2808接收索引。在至少一个实施例中,前端2808可以配置成确保在启动由传入命令缓冲区(例如,批处理缓冲区、推入缓冲区等)指定的工作负载之前,处理集群阵列2812配置成有效状态。

[0252] 在至少一个实施例中,并行处理单元2802的一个或多个实例中的每一个可以与并行处理器存储器2828耦合。在至少一个实施例中,可以由存储器交叉开关2816访问并行处理器存储器2828,所述存储器交叉开关2816可以接收来自处理集群阵列2812以及I/O单元2804的存储器请求。在至少一个实施例中,存储器交叉开关2816可以经由存储器接口2818访问并行处理器存储器2828。在至少一个实施例中,存储器接口2818可以包括多个分区单元(例如,分区单元2820A、分区单元2820B到分区单元2820N),其可各自耦合至并行处理器存储器2828的一部分(例如,存储单元)。在至少一个实施例中,多个分区单元2820A-2820N为配置为等于存储单元的数量,使得第一分区单元2820A具有对应的第一存储单元2824A,第二分区单元2820B具有对应的存储单元2824B,第N分区单元2820N具有对应的第N存储单元2824N。在至少一个实施例中,分区单元2820A-2820N的数量可以不等于存储单元的数量。

[0253] 在至少一个实施例中,存储单元2824A-2824N可以包括各种类型的存储器设

备,包括动态随机存取存储器 (DRAM) 或图形随机存取存储器,例如同步图形随机存取存储器 (SGRAM),包括图形双倍数据速率 (GDDR) 存储器。在至少一个实施例中,存储器单元 2824A-2824N还可 包括3D堆叠存储器,包括但不限于高带宽存储器 (HBM)。在至少一个 实施例中,可以跨存储器单元2824A-2824N来存储诸如帧缓冲区或纹理映 射的渲染目标,从而允许分区单元2820A-2820N并行地写入每个渲染目标 的部分,以有效地使用并行处理器存储器2828的可用带宽。在至少一个实 施例中,可以排除并行处理器存储器2828的本地实例,以有利于利用系统 存储器与本地高速缓存存储器结合的统一存储器设计。

[0254] 在至少一个实施例中,处理集群阵列2812的集群2814A-2814N中的 任何一个都可以处理将被写入并行处理器存储器2828内的任何存储器单 元2824A-2824N中的数据。在至少一个实施例中,存储器交叉开关2816 可以配置为将每个集群2814A-2814N的输出传输到任何分区单元 2820A-2820N或另一个集群2814A-2814N,集群2814A-2814N可以对输出 执行其他处理操作。在至少一个实施例中,每个集群2814A-2814N可以通 过存储器交叉开关 2816与存储器接口2818通信,以从各种外部存储设备 读取或写入各种外部存储设备。在至少一个实施例中,存储器交叉开关 2816具有到存储器接口2818的连接以及与I/O单元2804通 信,以及到并行 处理器存储器2828的本地实例的连接,从而使不同处理集群2814A-2814N 内的处理单元与系统存储器或不是并行处理单元2802本地的其他存储器 进行通信。在至少一个实施例中,存储器交叉开关2816可以使用虚拟通道 来分离集群2814A-2814N和分区 单元2820A-2820N之间的业务流。

[0255] 在至少一个实施例中,可以在单个插入卡上提供并行处理单元2802 的多个实例,或者可以将多个插入卡互连。在至少一个实施例中,并行处 理单元2802的不同实例可以配置成相互操作,即使不同实例具有不同数量 的处理核心,不同数量的本地并行处理器存储器 和/或其他配置差异。例如,在至少一个实施例中,并行处理单元2802的一些实例可以包 括相对于其他 实例而言更高精度的浮点单元。在至少一个实施例中,结合并行处理单元 2802或并行处理器2800的一个或更多个实例的系统可以以各种配置和形 式因素来实现,包括但不限于台式机、膝上型计算机或手持式个人计算机、服务器、工作站、游戏机和/或 嵌入式系统。

[0256] 图29是根据至少一个实施例的分区单元2920的框图。在至少一个 实施例中,分区 单元2920是图28的分区单元2820A-2820N之一的实例。在至少一个实施例中,分区单元 2920包括L2高速缓存2921、帧缓冲区接 口2925和光栅操作单元 (“ROP”) 2926。L2高速缓存 2921是读/写高 速缓存,其配置成执行从存储器交叉开关2916和ROP 2926接收的加载和 存储操作。在至少一个实施例中,L2高速缓存2921将读取未命中和紧急 回写请求输出到帧 缓冲区接口2925以进行处理。在至少一个实施例中,还 可以经由帧缓冲区接口2925将更新 发送到帧缓冲区以进行处理。在至少一 个实施例中,帧缓冲区接口2925与并行处理器存储器 中的存储器单元 (诸 如图28的存储器单元2824A-2824N (例如,在并行处理器存储器2828 内)) 之一相互作用。

[0257] 在至少一个实施例中,ROP 2926是一种处理单元,其执行光栅操作, 诸如模版、z 测试、混合等。在至少一个实施例中,ROP 2926然后输出存 储在图形存储器中的处理后的 图形数据。在至少一个实施例中,ROP 2926 包括压缩逻辑以压缩被写入存储器的深度或颜 色数据并解压缩从存储器读 取的深度或颜色数据。压缩逻辑可以是利用多种压缩算法中

的一种或更多种的无损压缩逻辑。在至少一个实施例中，ROP 2926执行的压缩的类型可以基于要压缩的数据的统计特性而变化。例如，在至少一个实施例中，基于每图块基础上的深度和颜色数据执行增量颜色压缩。

[0258] 在至少一个实施例中，ROP 2926包括在每个处理集群内（例如，图 28的集群 2814A-2814N），而不是在分区单元2920内。在至少一个实施例中，通过存储器交叉开关 2916而不是像素片段数据传输对像素数据的读取和写入请求。在至少一个实施例中，经处理的图形数据可以在显示设备上（诸如图27的一个或多个显示设备2710A-2710B之一）显示，由处理器2702路由以供进一步处理，或者由图28的并行处理器2800内的处理实体之一路由以供进一步处理。

[0259] 图30是根据至少一个实施例的并行处理单元内的处理集群3014的框图。在至少一个实施例中，处理集群是图28的处理集群2814A-2814N之一的实例。在至少一个实施例中，图28的一个或多个处理集群 2814A-2814N中的一个或多个可以配置成并行执行许多线程，其中“线程”是指在特定的一组输入数据上执行的特定程序的实例。在至少一个实施例中，单指令多数据 (SIMD) 指令发布技术用于支持大量线程的并行执行而无需提供多个独立的指令单元。在至少一个实施例中，使用单指令多线程 (SIMT) 技术来支持并行执行大量一般同步的线程，这使用了公共指令单元，该公共指令单元配置成向每个处理集群内的一组处理引擎发出指令。

[0260] 在至少一个实施例中，可以通过将处理任务分配给SIMT并行处理器的管线管理器3032来控制处理集群3014的操作。在至少一个实施例中，管线管理器3032从图28的调度器2810（例如，图30的调度器3010）接收指令，通过图形多处理器3034和/或纹理单元3036管理这些指令的执行。在至少一个实施例中，图形多处理器3034是SIMT并行处理器的示例性实例。然而，在至少一个实施例中，处理集群3014内可以包括不同架构的各种类型的SIMT并行处理器。在至少一个实施例中，在处理集群3014内可以包括图形多处理器3034的一个或多个实例。在至少一个实施例中，图形多处理器3034可以处理数据，并且数据交叉开关3040可以用于将处理后的数据分发到多个可能的目的（包括其他着色器单元）地之一。在至少一个实施例中，管线管理器3032可以通过指定要针对数据交叉开关3040分配的处理数据的目的地来促进处理数据的分配。

[0261] 在至少一个实施例中，处理集群3014内的每个图形多处理器3034可以包括相同的一组功能执行逻辑（例如，算术逻辑单元、负载存储单元等）。在至少一个实施例中，可以以管线方式配置功能执行逻辑，其中可以在先前的指令完成之前发出新的指令。在至少一个实施例中，功能执行逻辑支持多种操作，包括整数和浮点算术、比较操作、布尔运算、移位和各种代数函数的计算。在至少一个实施例中，可以利用相同的功能单元硬件来执行不同的操作，并且可以存在功能单元的任何组合。

[0262] 在至少一个实施例中，传送到处理集群3014的指令构成线程。在至少一个实施例中，跨一组并行处理引擎执行的一组线程是线程组。在至少一个实施例中，线程组在不同的输入数据上执行程序。在至少一个实施例中，线程组内的每个线程可被分配给图形多处理器3034内的不同处理引擎。在至少一个实施例中，线程组可包括比图形多处理器3034内的多个处理引擎更少的线程。在至少一个实施例中，当线程组包括的线程数少于处理引擎的数量时，一个或多个处理引擎在正在处理该线程组的循环期间可能是空闲的。在至

少一个实施例中,线程组还可以包括比图形多处理器3034 内的多个处理引擎更多的线程。在至少一个实施例中,当线程组包括比图形多处理器3034内的处理引擎更多的线程时,可以在连续的时钟周期内执行处理。在至少一个实施例中,可以在图形多处理器3034上同时执行多个线程组。

[0263] 在至少一个实施例中,图形多处理器3034包括内部高速缓存存储器,以执行加载和存储操作。在至少一个实施例中,图形多处理器3034可以放弃内部高速缓存并使用处理集群3014内的高速缓存存储器(例如,L1高速缓存3048)。在至少一个实施例中,每个图形多处理器3034还可以访问分区单元(例如,图28的分区单元2820A-2820N)内的L2高速缓存,这些分区单元在所有处理集群3014之间共享并且可以用于在线程之间传输数据。在至少一个实施例中,图形多处理器3034还可以访问片外全局存储器,其可以包括本地并行处理器存储器和/或系统存储器中的一个或多个。在至少一个实施例中,图28的并行处理单元2802外部的任何存储器都可以用作全局存储器。在至少一个实施例中,处理集群3014包括图形多处理器3034的多个实例并共享可以存储在L1高速缓存3048中的公共指令和数据。

[0264] 在至少一个实施例中,每个处理集群3014可以包括配置成将虚拟地址映射为物理地址的存储器管理单元(“MMU”)3045。在至少一个实施例中,MMU 3045的一个或多个实例可以驻留在图28的存储器接口2818内。在至少一个实施例中,MMU 3045包括一组页表条目(PTE),其用于将虚拟地址映射到图块的物理地址以及可选地映射到高速缓存存储器行。在至少一个实施例中,MMU 3045可以包括地址转换后备缓冲区(TLB)或可以驻留在图形多处理器3034或L1高速缓存3048或处理集群3014内的高速缓存。在至少一个实施例中,处理物理地址以分配表面数据访问局部性,以便在分区单元之间进行有效的请求交织。在至少一个实施例中,高速缓存行索引可以用于确定对高速缓存行的请求是命中还是未命中。

[0265] 在至少一个实施例中,可以配置处理集群3014,使得每个图形多处理器3034耦合到纹理单元3036,以执行纹理映射操作,例如,确定纹理样本位置、读取纹理数据以及过滤纹理数据。在至少一个实施例中,根据需要从内部纹理L1高速缓存(未示出)或从图形多处理器3034内的L1高速缓存中读取纹理数据,并从L2高速缓存、本地并行处理器存储器或系统存储器中获取纹理数据。在至少一个实施例中,每个图形多处理器3034将处理后的任务输出到数据交叉开关3040,以将处理后的任务提供给另一处理集群3014以进行进一步处理或将处理后的任务存储在L2高速缓存、本地并行处理器存储器、或经由存储器交叉开关3016的系统存储器中。在至少一个实施例中,preROP 3042(光栅前操作单元)配置成从图形多处理器3034接收数据,将数据引导至ROP单元,该ROP单元可以与本文所述的分区单元(例如,图28的分区单元2820A-2820N)一起定位。在至少一个实施例中,PreROP 3042单元可以执行用于颜色混合的优化、组织像素颜色数据以及执行地址转换。

[0266] 推理和/或训练逻辑1015用于执行与一个或多个实施例相关的推理和/或训练操作。本文结合图9和/或图10提供关于推理和/或训练逻辑1015的细节。在至少一个实施例中,推理和/或训练逻辑1015可以用于推理或预测软件程序以使用优化的决策树解决问题。在至少一个实施例中,推理和/或训练逻辑1015可以用于选择用于解决特定操作的内核。

[0267] 图31示出了根据至少一个实施例的图形多处理器3134。在至少一个实施例中,图形多处理器3134与图30中示出的处理集群3014的管线管理器3132耦合。在至少一个实施例中,图形多处理器3134具有执行管线,该执行管线包括但不限于指令高速缓存3152、指令单元3154、地址映射单元3156、寄存器文件3158、一个或更多个通用图形处理单元(GPGPU)核心3162和一个或更多个加载/存储单元3166。GPGPU核心3162和加载/存储单元3166与高速缓存存储器3172和共享存储器3170通过存储器和高速缓存互连3168耦合。

[0268] 在至少一个实施例中,指令高速缓存3152从管线管理器3132接收要执行的指令流。在至少一个实施例中,将指令高速缓存在指令高速缓存3152中并将其分派以供指令单元3154执行。在一个实施例中,指令单元3154可以分派指令作为线程组(例如,线程束),将每个线程组分配给GPGPU核心3162内的不同执行单元。在至少一个实施例中,指令可以通过在统一地址空间内指定地址来访问任何本地、共享或全局地址空间。在至少一个实施例中,地址映射单元3156可以用于将统一地址空间中的地址转换成可以由加载/存储单元3166访问的不同的存储器地址。

[0269] 在至少一个实施例中,寄存器文件3158为图形多处理器3134的功能单元提供了一组寄存器。在至少一个实施例中,寄存器文件3158为连接到图形多处理器3134的功能单元(例如,GPGPU核心3162、加载/存储单元3166)的数据路径的操作数提供了临时存储。在至少一个实施例中,在每个功能单元之间划分寄存器文件3158,使得为每个功能单元分配寄存器文件3158的专用部分。在至少一个实施例中,寄存器文件3158在图形多处理器3134正在执行的不同线程束之间划分。

[0270] 在至少一个实施例中,GPGPU核心3162可以各自包括用于执行图形多处理器3134的指令的浮点单元(FPU)和/或整数算术逻辑单元(ALU)。在至少一个实施例中,GPGPU核心3162在架构上可以相似或架构可能有所不同。在至少一个实施例中,GPGPU核心3162的第一部分包括单精度FPU和整数ALU,而GPGPU核心的第二部分包括双精度FPU。在至少一个实施例中,FPU可以实现用于浮点算法的IEEE 754-2008标准或启用可变精度浮点算法。在至少一个实施例中,图形多处理器3134可以另外包括一个或更多个固定功能或特殊功能单元,以执行特定功能,诸如复制矩形或像素混合操作。在至少一个实施例中,GPGPU核心中的一个或更多个也可以包括固定或特殊功能逻辑。

[0271] 在至少一个实施例中,GPGPU核心3162包括能够对多组数据执行单个指令的SIMD逻辑。在至少一个实施例中,GPGPU核心3162可以物理地执行SIMD4、SIMD8和SIMD16指令,并且在逻辑上执行SIMD1、SIMD2和SIMD32指令。在至少一个实施例中,用于GPGPU核心的SIMD指令可以在编译时由着色器编译器生成,或者在执行针对单程序多数据(SPMD)或SIMT架构编写和编译的程序时自动生成。在至少一个实施例中,可以通过单个SIMD指令来执行为SIMT执行模型配置的程序的多个线程。例如,在至少一个实施例中,可以通过单个SIMD8逻辑单元并行执行执行相同或相似操作的八个SIMT线程。

[0272] 在至少一个实施例中,存储器和高速缓存互连3168是将图形多处理器3134的每个功能单元连接到寄存器文件3158和共享存储器3170的互连网络。在至少一个实施例中,存储器和高速缓存互连3168是交叉开关互连,其允许加载/存储单元3166在共享存储器3170和寄存器文件3158之间实现加载和存储操作。在至少一个实施例中,寄存器文件3158可以以与GPGPU核心3162相同的频率操作,从而在GPGPU核心3162和寄存器文件3158之间

进行数据传输的延迟非常低。在至少一个实施例中,共享存储器3170可以用于启用在图形多处理器3134内的功能单元上执行的线程之间的通信。在至少一个实施例中,高速缓存存储器3172可以用作例如数据高速缓存,以高速缓存在功能单元和图30示出的纹理单元3036之间通信的纹理数据。在至少一个实施例中,共享存储器3170也可以用作程序管理的高速缓存。在至少一个实施例中,除了存储在高速缓存存储器3172中的自动高速缓存的数据之外,在GPGPU核心3162上执行的线程还可以以编程方式将数据存储在共享存储器中。

[0273] 在至少一个实施例中,如本文所述的并行处理器或GPGPU通信地耦合到主机/处理器核心,以加速图形操作、机器学习操作、图案分析操作以及各种通用GPU (GPGPU) 功能。在至少一个实施例中,GPU可以通过总线或其他互连(例如,诸如PCIe或NVLink的高速互连)通信地耦合到主机处理器/核心。在至少一个实施例中,GPU可以与核心集成在封装或芯片上,并通过内部处理器总线/互连(封装或芯片内部的)通信地耦合到核心。在至少一个实施例中,不管GPU连接的方式如何,处理器核心可以以工作描述符中包含的命令/指令序列的形式向GPU分配工作。在至少一个实施例中,GPU然后使用专用电路/逻辑来有效地处理这些命令/指令。

[0274] 推理和/或训练逻辑1015用于执行与一个或多个实施例相关的推理和/或训练操作。本文结合图9和/或图10提供关于推理和/或训练逻辑1015的细节。在至少一个实施例中,推理和/或训练逻辑1015可以用于推理或预测软件程序以使用优化的决策树解决问题。在至少一个实施例中,推理和/或训练逻辑1015可以用于选择用于解决特定操作的内核。

[0275] 图32示出了根据至少一个实施例的多GPU计算系统3200。在至少一个实施例中,多GPU计算系统3200可以包括经由主机接口开关3204耦合到多个通用图形处理单元(GPGPU) 3206A-D的处理器3202。在至少一个实施例中,主机接口开关3204是将处理器3202耦合到PCI快速(express)总线的PCI快速交换设备,处理器3202可以通过PCI快速总线与GPGPU 3206A-D通信。在至少一个实施例中,GPGPU 3206A-D可以由一组高速点对点GPU到GPU链路3216互连。在至少一个实施例中,GPU到GPU链路3216经由专用GPU链路连接到每个GPGPU 3206A-D。在至少一个实施例中,P2P GPU链路3216使得能够在每个GPGPU 3206A-D之间进行直接通信,而无需通过处理器3202所连接的主机接口总线3204进行通信。在至少一个实施例中,在GPU到GPU业务被定向到P2P GPU链路3216的情况下,主机接口总线3204保持可用于系统存储器访问或例如经由一个或多个网络设备与多GPU计算系统3200的其他实例通信。尽管在至少一个实施例中,GPGPU 3206A-D经由主机接口开关3204连接到处理器3202,但是在至少一个实施例中,处理器3202包括对P2P GPU链接3216的直接支持,并且可以直接连接到GPGPU 3206A-D。

[0276] 推理和/或训练逻辑1015用于执行与一个或多个实施例相关的推理和/或训练操作。本文结合图9和/或图10提供关于推理和/或训练逻辑1015的细节。在至少一个实施例中,推理和/或训练逻辑1015可以用于推理或预测软件程序以使用优化的决策树解决问题。在至少一个实施例中,推理和/或训练逻辑1015可以用于选择用于解决特定操作的内核。

[0277] 图33是根据至少一个实施例的图形处理器3300的框图。在至少一个实施例中,图

形处理器3300包括环形互连3302、管线前端3304、媒体引擎3337和图形核心3380A-3380N。在至少一个实施例中,环形互连3302将图形处理器3300耦合到其他处理单元,包括其他图形处理器或一个或更多个通用处理器核心。在至少一个实施例中,图形处理器3300是集成在多核处理系统内的许多处理器之一。

[0278] 在至少一个实施例中,图形处理器3300经由环形互连3302接收数批命令。在至少一个实施例中,输入命令由管线前端3304中的命令流化器3303解释。在至少一个实施例中,图形处理器3300包括可扩展执行逻辑,以经由图形核心3380A-3380N执行3D几何处理和媒体处理。在至少一个实施例中,对于3D几何处理命令,命令流化器3303将命令提供给几何管线3336。在至少一个实施例中,对于至少一些媒体处理命令,命令流化器3303将命令提供给视频前端3334,其耦合有媒体引擎3337。在至少一个实施例中,媒体引擎3337包括用于视频和图像后处理的视频质量引擎(VQE)3323,以及用于提供硬件加速媒体数据编码和解码的多格式编码/解码(MFX)3333引擎。在至少一个实施例中,几何管线3336和媒体引擎3337各自生成用于由至少一个图形核心3380A提供的线程执行资源的执行线程。

[0279] 在至少一个实施例中,图形处理器3300包括在模块化核心3380A-3380N(有时称为核心片)中起重要作用的可伸缩线程执行资源,每个模块化核心具有多个子核心3350A-3350N,3360A-3360N(有时称为核心子片)。在至少一个实施例中,图形处理器3300可以具有任意数量的图形核心3380A至3380N。在至少一个实施例中,图形处理器3300包括具有至少第一子核心3350A和第二子核心3360A的图形核心3380A。在至少一个实施例中,图形处理器3300是具有单个子核心(例如3350A)的低功率处理器。在至少一个实施例中,图形处理器3300包括多个图形核心3380A-3380N,每个图形核心包括一组第一子核心3350A-3350N和一组第二子核心3360A-3360N。在至少一个实施例中,第一子核心3350A-3350N中的每个子核心至少包括第一组执行单元3352A-3352N和媒体/纹理采样器3354A-3354N。在至少一个实施例中,第二子核心3360A-3360N中的每个子核心至少包括第二组执行单元3362A-3362N和采样器3364A-3364N。在至少一个实施例中,每个子核心3350A-3350N,3360A-3360N共享一组共享资源3370A-3370N。在至少一个实施例中,共享资源包括共享高速缓冲存储器和像素操作逻辑。

[0280] 推理和/或训练逻辑1015用于执行与一个或更多个实施例相关的推理和/或训练操作。本文结合图9和/或图10提供关于推理和/或训练逻辑1015的细节。在至少一个实施例中,推理和/或训练逻辑1015可以用于推理或预测软件程序以使用优化的决策树解决问题。在至少一个实施例中,推理和/或训练逻辑1015可以用于选择用于解决特定操作的内核。

[0281] 图34是示出根据至少一个实施例的用于处理器3400的微架构的框图,该处理器3400可以包括执行指令的逻辑电路。在至少一个实施例中,处理器3400可以执行指令,包括x86指令、ARM指令、用于专用集成电路(ASIC)的专用指令等。在至少一个实施例中,处理器3400可以包括用于存储封装数据的寄存器,例如作为加利福尼亚州圣克拉拉市英特尔公司采用MMX技术启用的微处理器中的64位宽MMX™寄存器。在至少一个实施例中,整数和浮点数形式可用的MMX寄存器可以与封装的数据元素一起运行,所述封装的数据元素伴随单指令多数据(“SIMD”)和流式SIMD扩展(“SSE”)指令。在至少一个实施例中,与SSE2、SSE3、SSE4、AVX或更高版本(一般称为“SSEx”)技术有关的128位宽XMM寄存器可以保存此

类封装数据操作数。在至少一个实施例中,处理器3400可以 执行指令以加速机器学习或深度学习算法、训练或推理。

[0282] 在至少一个实施例中,处理器3400包括有序前端(“前端”)3401,以提取要执行的指令并准备稍后在处理器管线中使用的指令。在至少一个 实施例中,前端3401可以包括几个单元。在至少一个实施例中,指令预取 器3426从存储器中获取指令并将指令提供给指令解码器3428,指令解码 器3428又对指令进行解码或解释。例如,在至少一个实施例中,指令解码 器3428将接收到的指令解码为机器可执行的所谓的“微指令”或“微操作”(也称为“微操作”或“微指令”)的一个或更多个操作。在至少一个实 施例中,指令解码器3428将指令解析为操作码以及相应的数据和控制字段,其可以由微架构用来使用以根据至少一个 实施例来执行操作。在至少一个 实施例中,追踪高速缓存3430可以将解码的微指令组装成微指令队列3434 中的程序排序的序列或追踪以供执行。在至少一个实施例中,当追踪高速 缓存3430遇到复杂指令时,微码ROM 3432提供完成操作所需的微指令。

[0283] 在至少一个实施例中,可以将一些指令转换成单个微操作,而另一 些指令则需要几个微操作来完成全部操作。在至少一个实施例中,如果需 要多于四个的微指令来完成一条指令,则指令解码器3428可以访问微码 ROM 3432以执行指令。在至少一个实施例中,可以将指令解码为少量的 微指令以在指令解码器3428处进行处理。在至少一个实施例中,如果需要 多个微指令完成操作,则可以将指令存储在微码ROM 3432中。在至少一 个实施例中,追踪高速缓存器3430参考入口点可编程逻辑阵列(“PLA”) 以确定正确的微指令指针,用于根据至少一个实施例从微码ROM 3432读 取微码序列以完成一个或更多个指令。在至少一个实施例中,在微码ROM 3432完成对指令的微操作排序之后,机器的前端3401可以恢 复从追踪高 速缓存3430获取微操作。

[0284] 在至少一个实施例中,乱序执行引擎(“乱序引擎”)3403可以准 备用于执行的指令。在至少一个实施例中,乱序执行逻辑具有多个缓冲区,以使指令流平滑并重新排序,以 在指令沿管线下落并被调度执行时优化性能。在至少一个实施例中,乱序执行引擎3403包括但不限于分配器/寄存 器重命名器3440、存储器微指令队列3442、整数/浮点微指令队列 3444、存储器调度器3446、快速调度器3402、慢速/通用浮点调度器(“慢速/通 用FP调度器”)3404和简单浮点调度器(“简单FP调度器”)3406。在 至少一个实施例中,快速调度器 3402、慢速/通用浮点调度器3404和简单 浮点调度器3406也统称为“微指令调度器3402、 3404、3406”。在至少 一个实施例中,分配器/寄存器重命名器3440分配每个微指令按顺序 执行 所需要的机器缓冲区和资源。在至少一个实施例中,分配器/寄存器重命名 器3440将逻辑寄存器重命名为寄存器文件中的条目。在至少一个实施例中, 分配器/寄存器重命名 器3440还为两个微指令队列之一中的每个微指令分 配条目,存储器微指令队列3442用于 存储器操作和整数/浮点微指令队列 3444用于非存储器操作,在存储器调度器3446和微指 令调度器3402、3404、3406的前面。在至少一个实施例中,微指令调度器3402、3404、3406基 于它们的从属输入寄存器操作数源的就绪性和需要完成的执行资源微指令 的可用性来确定何时准备好执行微指令。在至少一个实施例中,快速调度 器3402可以在主时钟周期的每个一半上调度,而慢速/通用浮点调度器 3404和简单浮点调度器3406可以在每个主处理器 时钟周期调度一次。在 至少一个实施例中,微指令调度器3402、3404、3406对调度端口进行 仲裁,以调度用于执行的微指令。

[0285] 在至少一个实施例中,执行块3411包括但不限于整数寄存器文件/支路网络3408、浮点寄存器文件/支路网络(“FP寄存器文件/支路网络”)3410、地址生成单元(“AGU”)3412和3414、快速算术逻辑单元(“快速ALU”)3416和3418、慢速算术逻辑单元(“慢速ALU”)3420、浮点ALU(“FP”)3422和浮点移动单元(“FP移动”)3424。在至少一个实施例中,整数寄存器文件/支路网络3408和浮点寄存器文件/支路网络3410在本文中也称为“寄存器文件3408、3410”。在至少一个实施例中,AGU 3412和3414、快速ALU 3416和3418、慢速ALU 3420、浮点ALU 3422 和浮点移动单元3424在本文中也称为“执行单元3412、3414、3416、3418、3420、3422和3424”。在至少一个实施例中,执行块3411可以包括但不限于任意数量(包括零)和类型的寄存器文件、支路网络、地址生成单元 和执行单元(以任何组合)。

[0286] 在至少一个实施例中,寄存器文件3408、3410可以布置在微指令调度器3402、3404、3406与执行单元3412、3414、3416、3418、3420、3422 和3424之间。在至少一个实施例中,整数寄存器文件/支路网络3408执行 整数运算。在至少一个实施例中,浮点寄存器文件/支路网络3410执行浮 点操作。在至少一个实施例中,寄存器文件3408、3410中的每一个可以包 括但不限于支路网络,该支路网络可以绕过或转发尚未写入寄存器文件中 的刚刚完成的结果到新的从属对象。在至少一个实施例中,寄存器文件 3408、3410可以彼此通信数据。在至少一个实施例中,整数寄存器文件/ 支路网络3408可以包括但不限于两个单独的寄存器文件、一个寄存器文件 用于低阶32位数据,第二寄存器文件用于高阶32位数据。在至少一个实 施例中,浮点寄存器文件/支路网络3410可以包括但不限于128位宽的条 目,因为浮点指令通常具有宽度为64至128位的操作数。

[0287] 在至少一个实施例中,执行单元3412、3414、3416、3418、3420、3422、3424可以执 行指令。在至少一个实施例中,寄存器文件3408、3410 存储微指令需要执行的整数和浮点 数据操作数值。在至少一个实施例中, 处理器3400可以包括但不限于任何数量的执行单元 3412、3414、3416、3418、3420、3422、3424及其组合。在至少一个实施例中,浮点ALU 3422 和浮点移动单元3424,可以执行浮点、MMX、SIMD、AVX和SSE或其 他操作,包括专门的机器学习指令。在至少一个实施例中,浮点ALU 3422 可以包括但不限于64位乘64位浮点除法器, 以执行除法、平方根和余数 微操作。在至少一个实施例中,可以用浮点硬件来处理涉及浮 点值的指令。在至少一个实施例中,可以将ALU操作传递给快速ALU 3416、3418。在 至少一个实施例中,快速ALUS 3416、3418可以以半个时钟周期的有效延 迟执行快速操作。在至少 一个实施例中,大多数复杂的整数运算进入慢速 ALU 3420,因为慢速ALU 3420可以包括但 不限于用于长延迟类型操作的 整数执行硬件,例如乘法器、移位、标志逻辑和分支处理。在 至少一个实 施例中,存储器加载/存储操作可以由AGUS 3412、3414执行。在至少一 个实施 例中,快速ALU 3416、快速ALU 3418和慢速ALU 3420可以对64 位数据操作数执行整数运 算。在至少一个实施例中,可以实现快速ALU 3416、快速ALU 3418和慢速ALU 3420以支持包 括十六、三十二、128、256等的各种数据位大小。在至少一个实施例中,浮点ALU 3422和浮 点移 动单元3424可以实现为支持具有各种宽度的位的一定范围的操作数。在至 少一个实 施例中,浮点ALU 3422和浮点移动单元3424可以结合SIMD和 多媒体指令对128位宽封装数 据操作数进行操作。

[0288] 在至少一个实施例中,微指令调度器3402、3404、3406在父加载完 成执行之前调

度从属操作。在至少一个实施例中,由于可以在处理器3400 中推测性地调度和执行微指令,处理器3400还可以包括用于处理存储器未命中的逻辑。在至少一个实施例中,如果数据高速缓存中的数据加载未命中,则可能存在在管线中正在运行的从属操作,其使调度器暂时没有正确的数据。在至少一个实施例中,一种重放机制追踪并重新执行使用不正确数据的指令。在至少一个实施例中,可能需要重放从属操作并且可以允许完成独立操作。在至少一个实施例中,处理器的至少一个实施例的调度器和重放机制也可以设计为捕获用于文本串比较操作的指令序列。

[0289] 在至少一个实施例中,术语“寄存器”可以指代可以用作识别操作数的指令的一部分的机载处理器存储位置。在至少一个实施例中,寄存器可以是那些可以从处理器外部使用的寄存器(从程序员的角度来看)。在至少一个实施例中,寄存器可能不限于特定类型的电路。相反,在至少一个实施例中,寄存器可以存储数据、提供数据并执行本文描述的功能。在至少一个实施例中,本文描述的寄存器可以通过处理器内的电路使用多种不同技术来实现,例如专用物理寄存器、使用寄存器重命名动态分配的物理寄存器、专用和动态分配的物理寄存器的组合等。在至少一个实施例中,整数寄存器存储32位整数数据。至少一个实施例的寄存器文件还包含八个用于封装数据的多媒体SIMD寄存器。

[0290] 推理和/或训练逻辑1015用于执行与一个或多个实施例相关的推理和/或训练操作。本文结合图9和/或图10提供关于推理和/或训练逻辑1015的细节。在至少一个实施例中,推理和/或训练逻辑1015可以用于推理或预测软件程序以使用优化的决策树解决问题。在至少一个实施例中,推理和/或训练逻辑1015可以用于选择用于解决特定操作的内核。

[0291] 图35示出了根据至少一个实施例的深度学习应用处理器3500。在至少一个实施例中,深度学习应用处理器3500使用指令,如果由深度学习应用处理器3500执行,则指令使深度学习应用处理器3500执行贯穿本公开描述的一些或全部过程和技术。在至少一个实施例中,深度学习应用处理器3500是专用集成电路(ASIC)。在至少一个实施例中,应用处理器3500 执行矩阵乘法运算或者“硬连线”到硬件中,作为执行一个或多个指令或两者的结果。在至少一个实施例中,深度学习应用处理器3500包括但不限于处理集群3510 (1) - 3510 (12)、芯片间链路(“ICL”) 3520 (1) - 3520 (12)、芯片间控制器(“ICC”) 3530 (1) - 3530 (2)、第二代高带宽存储器(“HBM2”) 3540 (1) - 3540 (4)、存储器控制器(“Mem Ctrlrs”) 3542 (1) - 3542 (4)、高带宽存储器物理层(“HBM PHY”) 3544 (1) - 3544 (4)、管理控制器中央处理单元(“管理控制器CPU”) 3550、外围组件互连快速控制器和直接存储器访问块(“PCIe 控制器和 DMA”) 3570、以及十六通道外围组件互连快速端口(“PCI Express x 16”) 3580。

[0292] 在至少一个实施例中,处理集群3510可以执行深度学习操作,包括基于一种或更多种训练技术计算的权重参数的推理或预测操作,包括本文所述的那些技术。在至少一个实施例中,每个处理集群3510可以包括但不限于任何数量和类型的处理器。在至少一个实施例中,深度学习应用处理器3500可以包括任何数量和类型的处理集群3500。在至少一个实施例中,芯片间链路3520是双向的。在至少一个实施例中,芯片间链路3520和芯片间控制器3530使多个深度学习应用处理器3500能够交换信息,包括从执行一个或多个神经网络中体现的一种或更多种机器学习算法而产生的激活信息。在至少一个实施例中,深度学习应用处理器3500可以包括任意数量(包括零)和类型的ICL 3520和ICC 3530。

[0293] 在至少一个实施例中，HBM2 3540提供总共32GB的存储器。HBM2 3540(i)与存储器控制器3542(i)和HBM PHY 3544(i)都相关联。在至少一个实施例中，任何数量的HBM2 3540可以提供任何类型和总量的高带宽存储器，并且可以与任何数量(包括零)和类型的存储器控制器3542和HBM PHY 3544相关联。在至少一个实施例中，可以用任何数量和类型的块替换SPI、I2C、GPIO 3560、PCIe控制器和DMA 3570和/或PCIe3580，以任何技术上可行的方式实现任何数量和类型的通信标准。

[0294] 推理和/或训练逻辑1015用于执行与一个或多个实施例相关的推理和/或训练操作。本文结合图9和/或图10提供关于推理和/或训练逻辑 1015的细节。在至少一个实施例中，推理和/或训练逻辑1015可以用于推理或预测软件程序以使用优化的决策树解决问题。在至少一个实施例中，推理和/或训练逻辑1015可以用于选择用于解决特定操作的内核。

[0295] 图36是根据至少一个实施例的神经形态处理器3600的框图。在至少一个实施例中，神经形态处理器3600可以从神经形态处理器3600外部的源接收一个或多个输入。在至少一个实施例中，这些输入可以被传输到神经形态处理器3600内的一个或多个神经元3602。在至少一个实施例中，可以使用包括一个或多个算术逻辑单元(ALU)的电路或逻辑来实现神经元3602及其组件。在至少一个实施例中，神经形态处理器3600可以包括但不限于成千上万个神经元3602的实例，但是可以使用任何合适数量的神经元3602。在至少一个实施例中，神经元3602的每个实例可以包括神经元输入3604和神经元输出3606。在至少一个实施例中，神经元 3602可以生成可以传输到神经元3602的其他实例的输入的输入。在至少一个实施例中，神经元输入3604和神经元输出3606可以经由突触3608互连。

[0296] 在至少一个实施例中，神经元3602和突触3608可以互连，使得神经形态处理器3600操作以处理或分析由神经形态处理器3600接收的信息。在至少一个实施例中，当通过神经元输入3604接收到的输入超过阈值时，神经元3602可以发送输出脉冲(或“触发”或“峰值”)。在至少一个实施例中，神经元3602可以对在神经元输入3604处接收到的信号进行求和或积分。例如，在至少一个实施例中，神经元3602可以实现为有泄漏的积分-触发神经元，其中如果求和(称为“膜电位”)超过阈值，则神经元 3602可以使用诸如sigmoid或阈值函数的传递函数来产生输出(或“触发”)。在至少一个实施例中，泄漏的积分-触发神经元可以将神经输入3604处接收到的信号求和成膜电位，并且可以应用衰减因子(或泄漏)以减小膜电位。在至少一个实施例中，如果在神经元输入3604处接收到足够快以超过阈值的多个输入信号(即，在膜电势衰减得太低而不能触发之前)，则泄漏的积分-触发神经元可能会触发。在至少一个实施例中，神经元3602可以使用接收输入、将输入积分到膜电位、并衰减膜电位的电路或逻辑来实现。在至少一个实施例中，可以对输入求平均，或者可以使用任何其他合适的传递函数。此外，在至少一个实施例中，神经元3602可以包括但不限于当将传递函数应用于神经元输入3604的结果超过阈值时在神经元输出3606处产生输出尖峰的比较器电路或逻辑。在至少一个实施例中，一旦神经元3602触发，它可以通过例如将膜电位复位为0或另一合适的默认值来忽略先前接收的输入信息。在至少一个实施例中，一旦膜电位被重置为0，则神经元3602可以在合适的时间段(或修复期)之后恢复正常操作。

[0297] 在至少一个实施例中，神经元3602可以通过突触3608互连。在至少一个实施例中

中,突触3608可以操作以将从第一神经元3602的输出的信号传输到第二神经元3602的输入。在至少一个实施例中,神经元3602可以在一个以上的突触3608实例上传输信息。在至少一个实施例中,神经元输出3606的一个或更多个实例可以通过突触3608的实例连接到同一神经元3602中神经元输入3604的实例。在至少一个实施例中,相对于突触3608的那个实例,神经元3602的实例产生要在突触3608的实例上传输的输出可以被称为“突触前神经元”。在至少一个实施例中,相对于突触3608的实例,神经元3602的实例接收通过突触3608的实例传输的输入可以被称为“突触后神经元”。在至少一个实施例中,关于突触3608的各种实例,因为神经元3602的实例可以接收来自一个或更多个突触3608实例的输入,并且还可以通过一个或更多个突触3608实例传输输出,因此神经元3602的单个实例可以既是“突触前神经元”又是“突触后神经元”。

[0298] 在至少一个实施例中,神经元3602可以被组织成一层或更多层。在至少一个实施例中,神经元3602的每个实例可以具有一个神经元输出3606,该神经元输出3606可以通过一个或更多个突触3608扇出到一个或更多个神经元输入3604。在至少一个实施例中,第一层3610中的神经元3602的神经元输出3606可以连接到第二层3612中的神经元3602的神经元输入3604。在至少一个实施例中,层3610可以被称为“前馈层”。在至少一个实施例中,在第一层3610的实例中神经元3602的每个实例可以扇出到第二层3612中的神经元3602的每个实例。在至少一个实施例中,第一层3610可以被称为“完全连接的前馈层”。在至少一个实施例中,在第二层3612的每个实例中的神经元3602的每个实例扇出到少于在第三层3614中的神经元3602的所有实例。在至少一个实施例中,第二层3612可以被称为“稀疏连接的前馈层”。在至少一个实施例中,第二层3612中的神经元3602可以扇出到多个其他层中的神经元3602,也包括第二层3612中的神经元3602。在至少一个实施例中,第二层3612可以被称为“循环层”。在至少一个实施例中,神经形态处理器3600可以包括但不限于循环层和前馈层的任何合适的组合,包括但不限于稀疏连接的前馈层和完全连接的前馈层。

[0299] 在至少一个实施例中,神经形态处理器3600可以包括但不限于可重新配置的互连架构或专用硬连线互连,以将突触3608连接到神经元3602。在至少一个实施例中,神经形态处理器3600可以包括但不限于电路或逻辑,其根据神经网络拓扑结构和神经元扇入/扇出,允许根据需要分配给不同神经元3602。例如,在至少一个实施例中,可以使用互连结构(诸如片上网络)或通过专用连接将突触3608连接到神经元3602。在至少一个实施例中,可以使用电路或逻辑来实现突触互连及其组件。在至少一个实施例中,推理和/或训练逻辑1015可以用于基于端点来执行分段。

[0300] 图37是根据至少一个实施例的处理系统的框图。在至少一个实施例中,系统3700包括一个或更多个处理器3702和一个或更多个图形处理器3708,并且可以是单处理器台式机系统,多处理器工作站系统或具有大量处理器3702或处理器核心3707的服务器系统。在至少一个实施例中,系统3700是结合在片上系统(SoC)集成电路内的处理平台,用于在移动、手持或嵌入式设备中使用。

[0301] 在至少一个实施例中,系统3700可以包括或结合在基于服务器的游戏平台中,包括游戏和媒体控制台、移动游戏控制台、手持游戏控制台或在线游戏控制台的游戏控制台。在至少一个实施例中,系统3700是移动电话、智能电话、平板计算设备或移动互联网设备。在至少一个实施例中,处理系统3700还可包括可穿戴设备,与可穿戴设备耦合或集成

在可穿戴设备中,可穿戴设备例如智能手表可穿戴设备、智能眼镜设备、增强现实设备或虚拟现实设备。在至少一个实施例中,处理系统3700是电视或机顶盒设备,其具有一个或更多个处理器3702以及由一个或更多个图形处理器 3708生成的图形界面。

[0302] 在至少一个实施例中,一个或更多个处理器3702各自包括一个或更多个处理器核心3707,以处理指令,这些指令在被执行时执行针对系统和用户软件的操作。在至少一个实施例中,一个或更多个处理器核心3707中的每一个被配置为处理特定指令集3709。在至少一个实施例中,指令集 3709可以促进复杂指令集计算(CISC)、精简指令集计算(RISC)、或通过超长指令字(VLIW)进行计算。在至少一个实施例中,处理器核心 3707每个可以处理不同的指令集3709,该指令集3709可以包括促进对其他指令集进行仿真的指令。在至少一个实施例中,处理器核心3707还可以包括其他处理设备,例如数字信号处理器(DSP)。

[0303] 在至少一个实施例中,处理器3702包括高速缓冲存储器3704。在至少一个实施例中,处理器3702可以具有单个内部高速缓存或多个级别的内部高速缓存。在至少一个实施例中,高速缓冲存储器在处理器3702的各个组件之间共享。在至少一个实施例中,处理器3702还使用外部高速缓存(例如,三级(L3)高速缓存或最后一级高速缓存(LLC))(未示出),外部高速缓存可以使用已知的高速缓存一致性技术在处理器内核3707之间共享该处理器。在至少一个实施例中,寄存器文件3706另外包括在处理器 3702中,该处理器文件可以包括用于存储不同类型的数据的不同类型的寄存器(例如,整数寄存器、浮点寄存器、状态寄存器和指令指针寄存器)。

[0304] 在至少一个实施例中,寄存器堆3706可以包括通用寄存器或其他寄存器。

[0305] 在至少一个实施例中,一个或更多个处理器3702与一个或更多个接口总线3710耦合,以在处理器3702与系统3700中的其他组件之间传输通信信号,例如地址、数据或控制信号。在至少一个实施例中,接口总线3710可以是处理器总线,例如直接媒体接口(DMI)总线的版本。在至少一个实施例中,接口3710不限于DMI总线,并且可以包括一个或更多个外围组件互连总线(例如,PCI、PCI快速)、存储器总线或其他类型的接口总线。在至少一个实施例中,处理器3702包括集成存储器控制器3716和平台控制器集线器3730。在至少一个实施例中,存储器控制器3716促进存储器设备与系统3700的其他组件之间的通信,而平台控制器集线器(PCH) 3730通过本地I/O总线提供到I/O设备的连接。

[0306] 在至少一个实施例中,存储器设备3720可以是动态随机存取存储器(DRAM)设备、静态随机存取存储器(SRAM)设备、闪存设备、相变存储器设备或具有适当的性能的一切其他存储器设备以用作进程存储器。

[0307] 在至少一个实施例中,存储器设备3720可以用作系统3700的系统存储器,以存储在一个或多个处理器3702执行应用或处理时使用的数据3722和指令3721。在至少一个实施例中,存储器控制器3716还与可选的外部图形处理器3712耦合,该外部图形处理器3712可以与处理器3702中的一个或更多个图形处理器3708通信以执行图形和媒体操作。在至少一个实施例中,显示设备3711可以连接至处理器3702。在至少一个实施例中,显示设备3711可以包括例如在移动电子设备或膝上型计算机设备或便携式计算机中的内部显示设备或通过显示接口(例如DisplayPort等)连接的外部显示设备中的一个或更多个。在至少一个实施例中,显示设备3711可以包括头戴式显示器(HMD),诸如用于虚拟现实(VR)应用

或增强现实 (AR) 应用的立体显示设备。

[0308] 在至少一个实施例中,平台控制器集线器3730使外围设备能够通过 高速I/O总线连接到存储设备3720和处理器3702。在至少一个实施例中, I/O外围设备包括但不限于音频控制器3746、网络控制器3734、固件接口 3728、无线收发器3726、触摸传感器3725、数据存储设备3724(例如, 硬盘驱动器、闪存等)。在至少一个实施例中,数据存储设备3724可以经由存储接口(例如,SATA)或经由诸如外围部件互连总线(例如,PCI,PCI快速)的外围总线来连接。在至少一个实施例中,触摸传感器3725 可以包括触摸屏传感器、压力传感器或指纹传感器。在至少一个实施例中, 无线收发器3726可以是Wi-Fi收发器、蓝牙收发器或诸如3G、4G或长期 演进(LTE)收发器的移动网络收发器。在至少一个实施例中,固件接口3728使能与系统固件的通信,并且可以是例如统一的可扩展固件接口 (UEFI)。在至少一个实施例中,网络控制器3734可以启用到有线网络 的网络连接。在至少一个实施例中,高性能网络控制器(未示出)与接口 总线3710耦合。在至少一个实施例中,音频控制器3746是多通道高清晰度音频控制器。在至少一个实施例中,系统3700包括可选的传统I/O控制 器3740,用于将传统(例如,个人系统2(PS/2))设备耦合到系统。在 至少一个实施例中,平台控制器集线器3730还可以连接到一个或更多个通 用串行总线(USB)控制器3742,一个或更多个通用串行总线(USB)控 制器3742连接输入设备,诸如键盘和鼠标3743组合、相机3744或其他 USB输入设备。

[0309] 在至少一个实施例中,可以将存储器控制器3716和平台控制器集线 器3730的实例集成到分立的外部图形处理器中,例如外部图形处理器3712。在至少一个实施例中,平 台控制器集线器3730和/或存储控制器3716可以 在一个或更多个处理器3702的外部。例如,在至少一个实施例中,系统 3700可以包括外部存储控制器3716和平台控制器集线器 3730,其可以被 配置为存储控制器集线器和与处理器3702通信的系统芯片组内的外围控 制器集线器。

[0310] 推理和/或训练逻辑1015用于执行与一个或更多个实施例相关的推 理和/或训练 操作。本文结合图9和/或图10提供关于推理和/或训练逻辑 1015的细节。在至少一个实施 例中,推理和/或训练逻辑1015可以用于推 理或预测软件程序以使用优化的决策树解决问题。在至少一个实施例中, 推理和/或训练逻辑1015可以用于选择用于解决特定操作的内 核。

[0311] 图38是根据至少一个实施例的具有一个或多个处理器核心 3802A-3802N,集成存 储器控制器3814和集成图形处理器3808的处理器 3800的框图。在至少一个实施例中,处理 器3800可包括相当于和包括由 虚线框表示的附加核心3802N的附加核心。在至少一个实施 例中,处理器 核心3802A-3802N中的每一个包括一个或更多个内部高速缓存单元 3804A-3804N。在至少一个实施例中,每个处理器核心还可以访问一个或 更多个共享高速缓存的 单元3806。

[0312] 在至少一个实施例中,内部高速缓存单元3804A-3804N和共享高速 缓存单元3806 表示处理器3800内的高速缓存存储器层次结构。在至少一 个实施例中,高速缓存存储器单 元3804A-3804N可以包括至少一级指令和 每个处理器核心和一个或更多个共享中级缓存 级别中的数据缓存,例如级 别2(L2)、级别3(L3)、级别4(L4)或其他级别的缓存,其中在外部 存储器之前的最高级别缓冲为归类为LLC。在至少一个实施例中,高速 缓存一致性逻辑

维持各种高速缓存单元3806和3804A-3804N之间的一致性。

[0313] 在至少一个实施例中,处理器3800还可包括一组一个或更多个总线 控制器单元3816和系统代理核心3810。在至少一个实施例中,一个或更 多个总线控制器单元3816管理一组外围总线,例如一个或更多个PCI或 PCI快速总线。在至少一个实施例中,系统代理核心3810为各种处理器组 件提供管理功能。在至少一个实施例中,系统代理核心3810包括一个或更 多个集成存储器控制器3814,以管理对各种外部存储器设备(未示出)的 访问。

[0314] 在至少一个实施例中,一个或更多个处理器核心3802A-3802N包括 对同时多线程的支持。在至少一个实施例中,系统代理核心3810包括用于 在多线程处理期间协调和操作核心3802A-3802N的组件。在至少一个实施 例中,系统代理核心3810可以另外包括功率控制单元(PCU),该功率控 制单元包括逻辑和组件以调节处理器核心3802A-3802N和图形处 理器 3808的一个或更多个功率状态。

[0315] 在至少一个实施例中,处理器3800另外包括图形处理器3808以执 行图形处理操作。在至少一个实施例中,图形处理器3808与共享高速缓存 单元3806和包括一个或更多个集成存储器控制器3814的系统代理核心 3810耦合。在至少一个实施例中,系统代理核心 3810还包括用于驱动图 形处理器输出到一个或多个耦合的显示器的显示控制器3811。在 至少一个 实施例中,显示控制器3811也可以是经由至少一个互连与图形处理器3808 耦合的独立模块,或者可以集成在图形处理器3808内。

[0316] 在至少一个实施例中,基于环的互连单元3812用于耦合处理器3800 的内部组件。在至少一个实施例中,可以使用替代性互连单元,例如点对 点互连、交换互连或其他技术。在至少一个实施例中,图形处理器3808 经由I/O链路3813与环形互连3812耦合。

[0317] 在至少一个实施例中,I/O链路3813代表多种I/O互连中的至少一种, 包括便于各种处理器组件与高性能嵌入式存储器模块3818(例如eDRAM 模块)之间的通信的封装上I/O互连。在至少一个实施例中,处理器核心 3802A-3802N和图形处理器3808中的每一个使用嵌入式存储器模块3818 作为共享的最后一级高速缓存。

[0318] 在至少一个实施例中,处理器核心3802A-3802N是执行公共指令集 架构的同质核 心。在至少一个实施例中,处理器核心3802A-3802N在指令 集架构(ISA)方面是异构的,其中 一个或更多个处理器核心3802A-3802N 执行一个公共指令集,而一个或更多个其他处 理器核心3802A-3802N执行 公共指令集或不同指令集的子集。在至少一个实施例中,就微架 构而言,处理器核心3802A-3802N是异构的,其中具有相对较高功耗的一个或更多 个核心 与具有较低功耗的一个或更多个功率核心耦合。在至少一个实施 例 中,处理器3800可以被实现在一个或更多个芯片上或被实现为SoC集成 电路。

[0319] 推理和/或训练逻辑1015用于执行与一个或更多个实施例相关的推 理和/或训练 操作。本文结合图9和/或图10提供关于推理和/或训练逻辑 1015的细节。在至少一个实施 例中,推理和/或训练逻辑1015可以用于推 理或预测软件程序以使用优化的决策树解决问题。在至少一个实施例中,推理和/或训练逻辑1015可以用于选择用于解决特定操作的内 核。

[0320] 图39是根据本文所述的至少一个实施例的图形处理器核心3900的 硬件逻辑的框 图。在至少一个实施例中,图形处理器核心3900被包括在图 形核心阵列内。在至少一个实 施例中,图形处理器核心3900(有时称为核 心切片)可以是模块化图形处理器内的一个或

更多图形核心。在至少一个实施例中，图形处理器核心3900是一个图形核心切片的示例，并且本文所述的图形处理器可以基于目标功率和性能包络线包括多个图形核心切片。在至少一个实施例中，每个图形核心3900可以包括与多个子核心 3901A-3901F耦合的固定功能块3930，也称为子切片，其包括通用和固定功能逻辑的模块块。

[0321] 在至少一个实施例中，固定功能块3930包括几何/固定功能管线3936，例如，在较低性能和/或较低功率的图形处理器实施方式中，该几何/固定功能管线3936可以由图形处理器3900中的所有子核心共享。在至少一个实施例中，几何/固定功能管线3936包括3D固定功能管线、视频前端单元，线程产生器和线程分派器以及管理统一返回缓冲区的统一返回缓冲区管理器。

[0322] 在至少一个实施例中，固定功能块3930还包括图形SoC接口3937、图形微控制器3938和媒体管线3939。在固定的至少一个实施例中，图形 SoC接口3937提供了图形核心3900以及片上集成电路系统中的其他处理器核心之间的接口。在至少一个实施例中，图形微控制器3938是可编程子处理器，其可配置为管理图形处理器3900的各种功能，包括线程分派、调度和抢占。在至少一个实施例中，媒体管线3939包括有助于对包括图像和视频数据的多媒体数据进行解码、编码、预处理和/或后处理的逻辑。在至少一个实施例中，媒体管线3939经由对子核心3901-3901F内的计算或采样逻辑的请求来实现媒体操作。

[0323] 在至少一个实施例中，SoC接口3937使图形核心3900能够与通用应用处理器核心（例如，CPU）和/或SoC内的其他组件通信，包括存储器层次结构元素，诸如共享的最后一级高速缓存、系统RAM和/或嵌入式片上或封装DRAM。在至少一个实施例中，SoC接口3937还可以使得能够与SoC内的固定功能设备（例如，相机成像管线）进行通信，并且使得能够使用和/或实现可以在图形核心3900和SoC内部的CPU之间共享的全局存储器原子。在至少一个实施例中，SoC接口3937还可以实现用于图形核心3900的电源管理控制，并且启用图形核心3900的时钟域与SoC内的其他时钟域之间的接口。在至少一个实施例中，SoC接口3937使得能够从命令流转化器和全局线程分派器接收命令缓冲区，其配置为向图形处理器内的一个或多个图形核心中的每一个提供命令和指令。在至少一个实施例中，当要执行媒体操作时，可以将命令和指令分派给媒体管线3939，或者当要执行图形处理操作时，可以将其分配给几何形状和固定功能管线（例如，几何形状和固定功能管线3936、几何形状和固定功能管线3914）。

[0324] 在至少一个实施例中，图形微控制器3938可以配置为对图形核心 3900执行各种调度和管理任务。在至少一个实施例中，图形微控制器3938可以在子核心3901A-3901F中的执行单元(EU)阵列3902A-3902F、3904A-3904F内的各种图形并行引擎上执行图形和/或计算工作负载调度。在至少一个实施例中，在包括图形核心3900的SoC的CPU核心上执行的主机软件可以提交多个图形处理器门铃之一的工作负载，其调用适当的图形引擎上的调度操作。在至少一个实施例中，调度操作包括确定接下来要运行哪个工作负载、将工作负载提交给命令流转化器、抢先在引擎上运行的现有工作负载、监控工作负载的进度以及在工作负载完成时通知主机软件。在至少一个实施例中，图形微控制器3938还可以促进图形核心3900的低功率或空闲状态，从而为图形核心3900提供在图形核心3900内独立于操作系统和/或系统上的图形驱动程序软件的跨低功率状态转换的保存和恢复寄存器的能力。

[0325] 在至少一个实施例中，图形核心3900可以具有比所示的子核心 3901A-3901F多或

少达N个模块化子核心。对于每组N个子核心,在至少一个实施例中,图形核心3900还可以包括共享功能逻辑3910、共享和/或高速缓存存储器3912、几何/固定功能管线3914以及附加的固定功能逻辑3916以加速各种图形和计算处理操作。在至少一个实施例中,共享功能逻辑3910可以包括可由图形核心3900内的每个N个子核心共享的逻辑单元(例如,采样器、数学和/或线程间通信逻辑)。在至少一个实施例中,共享和/或缓存存储器3912可以是图形核心3900内的N个子核心3901A-3901F的最后一级高速缓存,并且还可以用作可由多个子核心访问的共享存储器。在至少一个实施例中,可以包括几何/固定功能管线3914来代替固定功能块3930内的几何/固定功能管线3936,并且可以包括相同或相似的逻辑单元。

[0326] 在至少一个实施例中,图形核心3900包括附加的固定功能逻辑3916,其可以包括供图形核心3900使用的各种固定功能加速逻辑。在至少一个实施例中,附加的固定功能逻辑3916包括用于仅位置着色中使用的附加的几何管线。在仅位置着色中,存在至少两个几何管线,而在几何和固定功能管线3916、3936内的完整几何管线和剔除管线中,其是可以包括在附加的固定功能逻辑3916中的附加几何管线。在至少一个实施例中,剔除管线是完整几何管线的修整版。在至少一个实施例中,完整管线和剔除管线可以执行应用程序的不同实例,每个实例具有单独的环境。在至少一个实施例中,仅位置着色可以隐藏被丢弃的三角形的长剔除运行,从而在某些情况下可以更早地完成着色。例如,在至少一个实施例中,附加固定功能逻辑3916中的剔除管线逻辑可以与主应用程序并行执行位置着色器,并且通常比完整管线更快地生成关键结果,因为剔除管线获取并遮蔽顶点的位置属性,无需执行光栅化和将像素渲染到帧缓冲区。在至少一个实施例中,剔除管线可以使用生成的临界结果来计算所有三角形的可见性信息,而与这些三角形是否被剔除无关。在至少一个实施例中,完整管线(在这种情况下可以称为重播管线)可以消耗可见性信息来跳过剔除的三角形以仅遮盖最终传递到光栅化阶段的可见三角形。

[0327] 在至少一个实施例中,附加的固定功能逻辑3916还可包括机器学习加速逻辑,例如固定功能矩阵乘法逻辑,用于实现包括用于机器学习训练或推理的优化。

[0328] 在至少一个实施例中,在每个图形子核心3901A-3901F内包括一组执行资源,其可用于响应于图形管线、媒体管线或着色器程序的请求来执行图形、媒体和计算操作。在至少一个实施例中,图形子核心3901A-3901F包括多个EU阵列3902A-3902F、3904A-3904F,线程分派和线程间通信(TD/IC)逻辑3903A-3903F,3D(例如,纹理)采样器3905A-3905F,媒体采样器3906A-3906F,着色器处理器3907A-3907F和共享本地存储器(SLM)3908A-3908F。在至少一个实施例中,EU阵列3902A-3902F、3904A-3904F每个都包含多个执行单元,这些执行单元是通用图形处理单元,能够为图形、媒体或计算操作提供服务,执行浮点和整数/定点逻辑运算,包括图形、媒体或计算着色器程序。在至少一个实施例中,TD/IC逻辑3903A-3903F为子核心内的执行单元执行本地线程分派和线程控制操作,并促进在子核心的执行单元上执行的线程之间的通信。在至少一个实施例中,3D采样器3905A-3905F可以与纹理或其他3D图形相关的数据读取到存储器中。在至少一个实施例中,3D采样器可以基于与给定纹理相关联的配置的采样状态和纹理格式来不同地读取纹理数据。在至少一个实施例中,媒体采样器3906A-3906F可以基于与媒体数据相关联的类型和格式来执行类似的读取操作。在至少一个实施例中,每个图形子核心3901A-3901F可以可替代地包括

统一的3D和媒体采样器。在至少一个实施例中,在每个子核心3901A-3901F内的执行单元上执行的线程可以利用每个子核心内的共享本地存储器3908A-3908F,以使在线程组内执行的线程能够使用片上存储器的公共池来执行。

[0329] 推理和/或训练逻辑1015用于执行与一个或多个实施例相关的推理和/或训练操作。本文结合图9和/或图10提供关于推理和/或训练逻辑1015的细节。在至少一个实施例中,推理和/或训练逻辑1015可以用于推理或预测软件程序以使用优化的决策树解决问题。在至少一个实施例中,推理和/或训练逻辑1015可以用于选择用于解决特定操作的内核。

[0330] 图40-图41示出了根据至少一个实施例的包括图形处理器核心的处理元件的阵列的线程执行逻辑4000。图40示出了至少一个实施例,其中使用了线程执行逻辑4000。图41示出了根据至少一个实施例的执行单元的示例性内部细节。

[0331] 如图40中所示,在至少一个实施例中,线程执行逻辑4000包括着色器处理器4002、线程分派器4004、指令高速缓存4006、包括多个执行单元4008A-4008N的可扩展执行单元阵列、采样器4010、数据高速缓存4012和数据端口4014。在至少一个实施例中,可缩放执行单元阵列可以例如基于工作负载的计算要求,通过启用或禁用一个或多个执行单元(例如,执行单元4008A,4008B,4008C,4008D,直到4008N-1和4008N中的任何一个)来动态缩放。在至少一个实施例中,可缩放执行单元通过链路到每个执行单元的互连结构互连。在至少一个实施例中,线程执行逻辑4000包括通过指令高速缓存4006、数据端口4014、采样器4010和执行单元4008A-4008N中的一个或多个到存储器(诸如系统存储器或高速缓存存储器)的一个或多个连接。在至少一个实施例中,每个执行单元(例如4008A)是独立的可编程通用计算单元,其能够执行多个同时的硬件线程,同时针对每个线程并行处理多个数据元素。在至少一个实施例中,执行单元4008A-4008N的阵列可缩放以包括任意数量的单独执行单元。

[0332] 在至少一个实施例中,执行单元4008A-4008N主要用于执行着色器程序。在至少一个实施例中,着色器处理器4002可以处理各种着色器程序并经由线程分派器4004来分派与着色器程序相关联的执行线程。在至少一个实施例中,线程分派器4004包括用于仲裁来自图形和媒体管线的线程初始化庆祝以及在执行单元4008A-4008N中的一个或多个执行单元上实例化请求的线程的逻辑。例如,在至少一个实施例中,几何管线可以将顶点、镶嵌或几何着色器分派到线程执行逻辑以进行处理。在至少一个实施例中,线程分派器4004还可以处理来自执行着色器程序的运行时线程产生请求。

[0333] 在至少一个实施例中,执行单元4008A-4008N支持一种指令集,该指令集包括对许多标准3D图形着色器指令的本机支持,从而使图形库(例如Direct 3D和OpenGL)中的着色器程序只需最少的翻译即可执行。在至少一个实施例中,执行单元支持顶点和几何处理(例如,顶点程序、几何程序、顶点着色器)、像素处理(例如,像素着色器、片段着色器)和通用处理(例如,计算和媒体着色器)。在至少一个实施例中,每个执行单元4008A-4008N包括一个或多个算术逻辑单元(ALU),能够执行多发单指令多数据(SIMD),并且多线程操作实现了高效的执行环境尽管有更高的延迟存储器访问。在至少一个实施例中,每个执行单元内的每个硬件线程具有专用的高带宽寄存器文件和相关的独立线程状态。在至少一个实施例中,执行是每个时钟到管线的多次发出,管线能够进行整数、单精度和双精度浮

点运算、SIMD分支功能、逻辑运算、先验运算和其他其他 运算。在至少一个实施例中,在等待来自存储器或共享功能之一的数据时, 执行单元4008A-4008N内的依赖性逻辑使等待线程休眠直到返回了所请求 的数据。在至少一个实施例中,当等待线程正在休眠时,硬件资源可以专 用于处理其他线程。例如,在至少一个实施例中,在与顶点着色器操作相 关联的延迟期间,执行单元可以对像素着色器、片段着色器或另一类型的 着色器程序(包括不同的顶点着色器)执行操作。

[0334] 在至少一个实施例中,执行单元4008A-4008N中的每一个执行单元 在数据元素的阵列上进行操作。在至少一个实施例中,多个数据元素是“执 行大小”或指令的通道数。在至少一个实施例中,执行通道是用于指令内 的数据元素访问、屏蔽和流控制的执行的逻辑单元。在至少一个实施例中, 多个通道可以独立于用于特定图形处理器的多个物理算术逻辑单元(ALU) 或浮点单元(FPU)。在至少一个实施例中,执行单元4008A-4008N支持 整数和浮点数据类型。

[0335] 在至少一个实施例中,执行单元指令集包括SIMD指令。在至少一 个实施例中,各种数据元素可以作为封装数据类型存储在寄存器中,并且 执行单元将基于元素的数据大小来处理各种元素。例如,在至少一个实施 例中,当对256位宽的向量进行操作时,将向量的256位存储在寄存器中, 并且执行单元对向量进行操作,作为四个单独的64位封装数据 元素(四字(QW)大小数据元素)、八个单独的32位封装数据元素(双字(DW) 大小数据元素)、十六个单独的16位封装数据元素(单词(W)大小数据 元素)或三十二个单独的8位数据元素(字节(B)大小的数据元素)。然而,在至少一个实施例中,不同的向量宽度和寄存器大小是可能的。

[0336] 在至少一个实施例中,一个或更多个执行单元可以被组合成具有共 同于融合EU的线程控制逻辑(4007A-4007N)的融合执行单元 4009A-4009N。在至少一个实施例中,可以将多个EU合并成一个EU组。在至少一个实施例中,融合EU组中的每个EU可以配置为执行单 独的 SIMD硬件线程。融合的EU组中的EU的数量可以根据各种实施例而变化。在至少一个 实施例中,每个EU可以执行各种SIMD宽度,包括但不限于 SIMD8、SIMD16和SIMD32。在至少 一个实施例中,每个融合图形执行单 元4009A-4009N包括至少两个执行单元。例如,在至少 一个实施例中,融 合执行单元4009A包括第一EU4008A、第二EU4008A以及第一EU4008A 和 第二EU4008A共有的线程控制逻辑4007A。在至少一个实施例中,线程 控制逻辑4007A控制 在融合图形执行单元4009A上执行的线程,从而允许 融合执行单元4009A-4009N内的每个 EU使用公共指令指针寄存器来执行。

[0337] 在至少一个实施例中,一个或更多个内部指令高速缓存(例如4006) 被包括在线程执行逻辑4000中以高速缓存用于执行单元的线程指令。在至 少一个实施例中,包括一个 或更多个数据高速缓存(例如4012)以在线程 执行期间高速缓存线程数据。在至少一个实 施例中,包括采样器4010以提 供用于3D操作的纹理采样和用于媒体操作的媒体采样。在至 少一个实施 例中,采样器4010包括专门的纹理或媒体采样功能,以在将采样数据提供 给 执行单元之前在采样过程中处理纹理或媒体数据。

[0338] 在执行期间,在至少一个实施例中,图形和媒体管线通过线程产生 和分派逻辑将 线程发起请求发送到线程执行逻辑4000。在至少一个实施例 中,一旦一组几何对象已经被 处理并光栅化成像素数据,则在着色器处理 器4002内的像素处理器逻辑(例如,像素着色

器逻辑、片段着色器逻辑等) 被调用以进一步计算输出信息并且导致将结果写入输出表面(例如,颜色 缓冲区、深度缓冲区、模板缓冲区等)。在至少一个实施例中,像素着色 器或片段着色器计算要在光栅化对象上插值的各种顶点属性的值。在至少 一个实施例中,着色器处理器4002内的像素处理器逻辑然后执行应用程序 接口(API)提供的像素或片段着色器程序。在至少一个实施例中,为了 执行着色器程序,着色器处理器4002经由线程分派器4004将线程分派到 执行单元(例如4008A)。在至少一个实施例中,着色器处理器4002使用 采样器4010中的纹理采样逻辑来访问存储在存储器中的纹理贴图纹理数据。在至少一个实施例中,对纹理数据和输入几何数据的算术运算为 每个几何片段计算像素颜色数据,或者丢弃一个或多个像素以进行进一步处理。

[0339] 在至少一个实施例中,数据端口4014提供了一种用于线程执行逻辑4000的存储器访问机制,以将处理后的数据输出到存储器以在图形处理器 输出管线上进行进一步处理。在至少一个实施例中,数据端口4014包括或 耦合到一个或多个高速缓存存储器(例如,数据高速缓存4012)以高速 缓存数据以便经由数据端口进行存储器访问。

[0340] 如图41所示,在至少一个实施例中,图形执行单元4008可以包括 指令获取单元4037、通用寄存器文件阵列(GRF) 4024、架构寄存器文件 阵列(ARF) 4026、线程仲裁器4022、发送单元4030、分支单元4040、一组SIMD浮点单元(FPU) 4034,以及在至少一个实施例中,一组专用 整数SIMD ALU4035。在至少一个实施例中,GRF4024和ARF4026包括 一组与可以在图形执行单元4008中活跃的每个同时硬件线程相关联的通用寄存器文件和架构寄存器文件。在至少一个实施例中,在ARF4026中维护每个线程架构状态,而在线程执行期间使用的数据存储GRF4024中。在至少一个实施例中,每个线程的执行状态,包括每个线程的指令指针,可以被保存在ARF4026中的线程专用寄存器中。

[0341] 在至少一个实施例中,图形执行单元4008具有一种架构,该架构是 同时多线程(SMT)和细粒度交错多线程(IMT)的组合。在至少一个实 施例中,架构具有模块化配置,该模块化配置可以在设计时基于同时线程 的目标数量和每个执行单元的寄存器数量来进行微调,其中执行单元资源 在用于执行多个同时线程的逻辑上分配。

[0342] 在至少一个实施例中,图形执行单元4008可以共同发布多个指令, 每个指令可以是不同的指令。在至少一个实施例中,图形执行单元线程 4008的线程仲裁器4022可以将指令分派到发送单元4030、分支单元4042 或SIMD FPU 4034之一以供执行。在至少一个实施例中,每个执行线程可 以访问GRF 4024中的128个通用寄存器,其中每个寄存器可以存储 32个 字节,可以作为32位数据元素的SIMD 8元素向量进行访问。在至少一个 实施例中,每个执行单元线程可以访问GRF 4024中的4KB,尽管实施例 不限于此,并且在其他实施例中可以提供更多或更少的寄存器资源。在至 少一个实施例中,尽管每个执行单元的线程数量也可以根据实施例而变化, 但是最多可以同时执行七个线程。在其中七个线程可以访问 4KB的至少 一个实施例中,GRF 4024可以存储总共28KB。在至少一个实施例中,灵 活的寻址模式可以允许将寄存器一起寻址以有效地建立更宽的寄存器或表 示跨步的矩形块数据结构。

[0343] 在至少一个实施例中,经由由消息传递发送单元4030执行的“发送” 指令来调度存储器操作、采样器操作和其他更长延迟的系统通信。在至少 一个实施例中,将分支指令分派到专用分支单元4032促进SIMD发散和最 终收敛。

[0344] 在至少一个实施例中,图形执行单元4008包括一个或更多个SIMD 浮点单元(FPU) 4034,以执行浮点操作。在至少一个实施例中,FPU 4034 还支持整数计算。在至少一个实施例中,FPU 4034可以SIMD执行多达M 个32位浮点(或整数)运算,或者SIMD执行多达2M个16位整数或16 位浮点运算。在至少一个实施例中,FPU中的至少一个提供扩展的数学能力以支持高吞吐量的先验数学函数和双精度64位浮点。在至少一个实施例中,还存在一组8位整数SIMD ALU 4035,并且可以被专门优化以执行与机器学习计算相关的操作。

[0345] 在至少一个实施例中,可以在图形子核心分组(例如,子切片)中 实例化图形执行单元4008的多个实例的阵列。在至少一个实施例中,执行 单元4008可以跨多个执行通道执行指令。在至少一个实施例中,在图形执 行单元4008上执行的每个线程在不同的通道上执行。

[0346] 推理和/或训练逻辑1015用于执行与一个或更多个实施例相关的推 理和/或训练操作。本文结合图9和/或图10提供关于推理和/或训练逻辑 1015的细节。在至少一个实施例中,推理和/或训练逻辑1015可以用于推 理或预测软件程序以使用优化的决策树解决问题。在至少一个实施例中, 推理和/或训练逻辑1015可以用于选择用于解决特定操作的内核。

[0347] 图42示出了根据至少一个实施例的并行处理单元(“PPU”) 4200。在至少一个实施例中,PPU 4200配置有机器可读代码,该机器可读代码如 果由PPU 4200执行,则使得PPU 4200执行贯穿本公开描述的一些或全部 过程和技术。在至少一个实施例中,PPU 4200是在一个或更多个集成电路 设备上实现的多线程处理器,并且利用多线程作为被设计为处理 在多个线 程上并行执行的计算机可读指令(也称为机器可读指令或简单的指令)的 延迟隐藏技术。在至少一个实施例中,线程是指执行线程,并且是被配置 为由PPU 4200执行的一组指令的实例。在至少一个实施例中,PPU 4200 是图形处理单元(“GPU”),图形处理单元配置为实现用于处理三维(“3D”) 图形数据的图形渲染管线,以便生成用于在显示设备(诸如液晶显示器 (“LCD”) 设备)上显示的二维(“2D”)图像数据。在至少一个实施 例中,PPU 4200用于执行计算,诸如线性代数运算和机器学习运算。图 42仅出于说明性目的示出了示例并行处理器,并且应被解释为在本公开的 范围内设想的处理器架构的非限制性示例,并且可以采用任何适当的处理 器来对其进行补充和/或替代。

[0348] 在至少一个实施例中,一个或更多个PPU 4200配置成加速高性能计 算(“HPC”)、数据中心和机器学习应用程序。在至少一个实施例中, PPU 4200配置成加速深度学习系统和应用程序,包括以下非限制性示例: 自动驾驶汽车平台、深度学习、高精度语音、图像、文本识别系统、智能 视频分析、分子模拟、药物发现、疾病诊断、天气预报、大数据分析、天 文学、分子动力学模拟、财务建模、机器人技术、工厂自动化、实时语言 翻译、在线搜索优化以及个性化用户推荐等。

[0349] 在至少一个实施例中,PPU 4200包括但不限于输入/输出(“I/O”) 单元4206、前端单元4210、调度器单元4212、工作分配单元4214、集线 器4216、交叉开关(“Xbar”) 4220、一个或更多个通用处理集群(“GPC”) 4218和一个或更多个分区单元(“内存分区单元”) 4222。在至少一个实 施例中,PPU 4200通过一个或更多个高速GPU互连(“GPU互连”) 4208 连接到主机处理器或其他PPU 4200。在至少一个实施例中,PPU 4200通 过互连4202连接到主机处理器或其他外围设备。在一实施例中,PPU 4200 连接到包括一个或更多个存储器设备(“存

储器”)4204的本地存储器。在至少一个实施例中,存储器设备4204包括但不限于一个或更多个动态随机存取存储器(“DRAM”)设备。在至少一个实施例中,一个或更多个DRAM设备配置和/或可配置为高带宽存储器(“HBM”)子系统,并且在每个设备内堆叠有多个DRAM管芯。

[0350] 在至少一个实施例中,高速GPU互连4208可以指代系统使用其来进行缩放的基于线的多通道通信链路,并包括与一个或更多个中央处理单元结合的一个或更多个PPU 4200(“CPU”),支持PPU 4200和CPU之间的缓存相干以及CPU主控。在至少一个实施例中,高速GPU互连4208通过集线器4216将数据和/或命令传输到PPU 4200的其他单元,例如一个或更多个复制引擎、视频编码器、视频解码器、电源管理单元和/或在图42中可能未明确示出的其他组件。

[0351] 在至少一个实施例中,I/O单元4206配置为通过系统总线4202从主机处理器(图42中未示出)发送和接收通信(例如,命令、数据)。在至少一个实施例中,I/O单元4206直接通过系统总线4202或通过一个或更多个中间设备(例如存储器桥)与主机处理器通信。在至少一个实施例中,I/O单元4206可以经由系统总线4202与一个或更多个其他处理器(例如一个或更多个PPU 4200)通信。在至少一个实施例中,I/O单元4206实现外围组件互连快速(“PCIe”)接口,用于通过PCIe总线进行通信。在至少一个实施例中,I/O单元4206实现用于与外部设备通信的接口。

[0352] 在至少一个实施例中,I/O单元4206对经由系统总线4202接收的分组进行解码。在至少一个实施例中,至少一些分组表示被配置为使PPU 4200执行各种操作的命令。在至少一个实施例中,I/O单元4206如命令所指定的那样将解码的命令发送到PPU 4200的各种其他单元。在至少一个实施例中,命令被发送到前端单元4210和/或被发送到集线器4216或PPU 4200的其他单元,例如一个或更多个复制引擎、视频编码器、视频解码器、电源管理单元等(图42中未明确示出)。在至少一个实施例中,I/O单元4206配置为在PPU 4200的各种逻辑单元之间路由通信。

[0353] 在至少一个实施例中,由主机处理器执行的程序在缓冲区中对命令流进行编码,该缓冲区将工作负载提供给PPU 4200以进行处理。在至少一个实施例中,工作负载包括指令和要由那些指令处理的数据。在至少一个实施例中,缓冲区是可由主机处理器和PPU 4200两者访问(例如,读/写)的存储器中的区域—主机接口单元可以配置为访问经由I/O单元4206通过系统总线4202传输的存储器请求连接到系统总线4202的系统存储器中的缓冲区。在至少一个实施例中,主机处理器将命令流写入缓冲区,然后将指示命令流开始的指针发送给PPU 4200,使得前端单元4210接收指向一个或更多个命令流指针并管理一个或更多个命令流,从命令流中读取命令并将命令转发到PPU 4200的各个单元。

[0354] 在至少一个实施例中,前端单元4210耦合到调度器单元4212,该调度器单元4212配置各种GPC 4218以处理由一个或更多个命令流定义的任务。在至少一个实施例中,调度器单元4212配置为追踪与调度器单元4212管理的各种任务有关的状态信息,其中状态信息可以指示任务被分配给哪个GPC 4218,任务是活跃的还是非活跃的,与任务相关联的优先级等等。在至少一个实施例中,调度器单元4212管理在一个或更多个GPC 4218上执行的多个任务。

[0355] 在至少一个实施例中,调度器单元4212耦合到工作分配单元4214,该工作分配单

元4214配置为分派任务以在GPC 4218上执行。在至少一个 实施例中,工作分配单元4214追踪从调度器单元4212接收到的多个调度 任务并且工作分配单元4214管理每个GPC 4218的待处理任务池和活跃任 务池。在至少一个实施例中,待处理任务池包括多个时隙(例如32个时隙), 这些时隙包含分配给要由特定的GPC 4218处理的任务;活跃任务池可包 括用于由GPC 4218主动处理的任务的多个时隙(例如4个时隙),以使 随着GPC 4218中的一个完成任务的执行,该任务将从GPC 4218的活动任 务池中逐出,并且从待处理任务池中选择其他任务之一,并安排其在GPC 4218上执行。在至少一个实施例中,如果活跃任务在GPC 4218上处于空 闲状态,例如在等待数据依赖性解决时,则活跃任务从GPC 4218中驱逐 并返回到待处理任务池,同时选择了待处理任务池中的另一个任务并调度 在GPC 4218上执行。

[0356] 在至少一个实施例中,工作分配单元4214经由XBar 4220与一个或 更多个GPC 4218通信。在至少一个实施例中,XBar 4220是互连网络,其 将PPU 4200的许多单元耦合到 PPU 4200的其他单元,并且可以配置为将 工作分配单元4214耦合到特定的GPC 4218。在至少一个实施例中,一个 或更多个PPU 4200的其他单元也可以通过集线器4216连接到XBar 4220。

[0357] 在至少一个实施例中,任务由调度器单元4212管理,并由工作分配 单元4214分配给GPC 4218之一。在至少一个实施例中,GPC 4218配置 为处理任务并产生结果。在至少一个实施例中,结果可以由GPC 4218中 的其他任务消耗,通过XBar 4220路由到不同的GPC 4218或存储在存储 器4204中。在至少一个实施例中,结果可以通过分区单元4222写到存储 器4204中,其实现了用于向存储器4204写入数据或从存储器4204读取数 据的存储器接口。在至少一个实施例中,结果可以经由高速GPU互连4208 传输到另一PPU 4204或CPU。在至少一个实施例中,PPU 4200包括但不 限于U个分区单元4222,其等于耦合到PPU 4200的分离且不同的存储设 备4204的数量。在至少一个实施例中,下面结合图44更详细地描述分区 单元4222。

[0358] 在至少一个实施例中,主机处理器执行驱动器核心,该驱动程序核 心实现应用程序编程接口(API),该应用程序编程接口使在主机处理器 上执行的一个或更多个应用程序能够调度操作以在PPU 4200上执行。在 一个实施例中,多个计算应用由PPU 4200同时执行,并且PPU 4200为多 个计算应用程序提供隔离、服务质量(“QoS”)和独立的地址空间。在 至少一个实施例中,应用程序生成指令(例如,以API调用的形式),该 指令使驱动器核心生成一个或更多个任务以供PPU 4200执行,并且驱动 器核心将任务输出至由PPU 4200处理的一个或更多个流。在至少一个实 施例中,每个任务包括一个或更多个相关线程组,其可以被称为线程束 (warp)。在至少一个实施例中,线程束包括可以并行执行的多个相关线 程(例如32个线程)。在至少一个实施例中,协作线程可以指代多个线程, 包括用于执行任务并且通过共享存储器交换数据的指令。在至少一个实 施例中,结合图44更详细地描述了线程和协作线程。

[0359] 推理和/或训练逻辑1015用于执行与一个或更多个实施例相关的推 理和/或训练操作。本文结合图9和/或图10提供关于推理和/或训练逻辑 1015的细节。在至少一个实 施例中,推理和/或训练逻辑1015可以用于推 理或预测软件程序以使用优化的决策树解决问题。在至少一个实施例中, 推理和/或训练逻辑1015可以用于选择用于解决特定操作的内 核。

[0360] 图43示出了根据至少一个实施例的通用处理集群(“GPC”)4300。在至少一个实施例中, GPC 4300是图40的GPC 4018。在至少一个实施例中, 每个GPC 4300包括但不限于用于处理任务的多个硬件单元, 并且每个GPC 4300包括但不限于管线管理器4302、预光栅操作单元(“PROP”) 4304、光栅引擎4308、工作分配交叉开关(“WDX”) 4316、存储器管理单元(“MMU”) 4318、一个或更多个数据处理集群(“DPC”) 4306, 以及部件的任何合适组合。

[0361] 在至少一个实施例中, GPC 4300的操作由管线管理器4302控制。在至少一个实施例中, 管线管理器4302管理一个或更多个DPC 4306的配置, 以处理分配给GPC 4300的任务。在至少一个实施例中, 管线管理器4302配置一个或更多个DPC 4306中的至少一个以实现图形渲染管线的至少一部分。在至少一个实施例中, DPC 4306配置为在可编程流式多处理器(“SM”) 4314上执行顶点着色器程序。在至少一个实施例中, 管线管理器4302配置为将从工作分配单元接收的数据包路由到GPC 4300内的适当逻辑单元, 以及在至少一个实施例中, 可以将一些数据包路由到PROP 4304和/或光栅引擎4308中的固定功能硬件单元, 而可以将其他数据包路由到DPC 4306以由图元引擎4312或SM 4314进行处理。在至少一个实施例中, 管线管理器4302配置DPC 4306中的至少一个以实现神经网络模型和/或计算管线。

[0362] 在至少一个实施例中, PROP单元4304配置为在至少一个实施例中将由光栅引擎4308和DPC 4306生成的数据路由到上面结合图42更详细地描述的分区单元4222中的光栅操作(“ROP”)单元。在至少一个实施例中, PROP单元4304配置为执行用于颜色混合的优化、组织像素数据、执行地址转换等等。在至少一个实施例中, 光栅引擎4308包括但不限于配置为执行各种光栅操作的多个固定功能硬件单元, 并且在至少一个实施例中, 光栅引擎4308包括但不限于设置引擎、粗光栅引擎、剔除引擎、裁剪引擎、精细光栅引擎、图块聚合引擎及其任意合适的组合。在至少一个实施例中, 设置引擎接收变换后的顶点并生成与由顶点定义的几何图元相关联的平面方程; 平面方程式被传送到粗光栅引擎以生成基本图元的覆盖信息(例如, 图块的x、y覆盖范围掩码); 粗光栅引擎的输出将传输到剔除引擎, 在剔除引擎中与z测试失败的图元相关联的片段将被剔除, 并传输到剪切引擎, 在剪切引擎中剪切位于视锥范围之外的片段。在至少一个实施例中, 将经过裁剪和剔除的片段传递给精细光栅引擎, 以基于设置引擎生成的平面方程式生成像素片段的属性。在至少一个实施例中, 光栅引擎4308的输出包括将由任何适当的实体(例如, 由在DPC 4306内实现的片段着色器)处理的片段。

[0363] 在至少一个实施例中, 包括在GPC 4300中的每个DPC 4306包括但不限于M管线控制器(“MPC”) 4310; 图元引擎4312; 一个或更多个SM 4314; 及其任何合适的组合。在至少一个实施例中, MPC 4310控制DPC 4306的操作, 将从管线管理器4302接收的分组路由到DPC 4306中的适当单元。在至少一个实施例中, 将与顶点相关联的分组路由到图元引擎4312, 图元引擎4312配置为从存储器中获取与顶点关联的顶点属性; 相反, 可以将与着色器程序相关联的数据包发送到SM 4314。

[0364] 在至少一个实施例中, SM 4314包括但不限于可编程流式处理器, 其配置为处理由多个线程表示的任务。在至少一个实施例中, SM 4314是多线程的并且配置为同时执行来自特定线程组的多个线程(例如32个线程), 并且实现单指令多数据(“SIMD”)架构, 其中将一组线程(例如, 线程束)中的每个线程配置为基于相同的指令集来处理不同的数据集。

在至少一个实施例中,线程组中的所有线程执行相同的指令。在至少一个实施例中,SM 4314实施单指令多线程(“SIMT”)架构,其中一组线程中的每个线程配置为基于相同的指令集来处理不同的数据集,但是其中线程组中的各个线程允许在执行期间发散。在至少一个实施例中,为每个线程束维护程序计数器、调用栈和执行状态,从而当线程束中的线程发散时,实现线程束和线程束内的串行执行之间的并发性。在另一个实施例中,为每个单独的线程维护程序计数器、调用栈和执行状态,从而使得在线程束内和线程束之间的所有线程之间具有相等的并发性。在至少一个实施例中,为每个单独的线程维持执行状态,并且可以收敛并并行地执行执行相同指令的线程以提高效率。下面更详细地描述SM 4314的至少一个实施例。

[0365] 在至少一个实施例中,MMU 4318在GPC 4300和内存分区单元(例如,图42的分区单元4222)之间提供接口,并且MMU 4318提供虚拟地址到物理地址的转换、存储器保护以及存储器请求的仲裁。在至少一个实施例中,MMU 4318提供一个或多个转换后备缓冲区(“TLB”),用于执行虚拟地址到存储器中的物理地址的转换。

[0366] 推理和/或训练逻辑1015用于执行与一个或多个实施例相关的推理和/或训练操作。本文结合图9和/或图10提供关于推理和/或训练逻辑1015的细节。在至少一个实施例中,推理和/或训练逻辑1015可以用于推理或预测软件程序以使用优化的决策树解决问题。在至少一个实施例中,推理和/或训练逻辑1015可以用于选择用于解决特定操作的内核。

[0367] 图44示出了根据一个实施例的并行处理单元(“PPU”)的内存分区单元4400。在至少一个实施例中,内存分区单元4400包括但不限于光栅操作(“ROP”)单元4402;二级(“L2”)高速缓存4404;存储器接口4406;及其任何合适的组合。在至少一个实施例中,存储器接口4406耦合到存储器。在至少一个实施例中,存储器接口4406可以实现32、64、128、1024位数据总线或类似的实现,用于高速数据传输。在至少一个实施例中,PPU包括U个存储器接口4406(其中U为正整数)、每对分区单元4400一个存储器接口4406,其中每对分区单元4400连接到相应的存储器设备。例如,在至少一个实施例中,PPU最多可以连接到Y个存储设备,例如高带宽存储堆栈或图形双数据速率、版本5、同步动态随机存取存储器(“GDDR5 SDRAM”)。

[0368] 在一个实施例中,存储器接口4406实现了高带宽存储器二代(“HBM2”)存储器接口,并且Y等于U的一半。在至少一个实施例中,HBM2存储器堆栈与PPU位于同一物理封装上,与传统的GDDR5 SDRAM系统相比,节省了大量的功率和面积。在一个实施例中,每个HBM2堆栈包括四个存储器管芯,并且Y=4,而HBM2堆栈包括每个管芯两个128位通道,总共8个通道和1024位的数据总线宽度。在至少一个实施例中,存储器支持单错误校正双错误检测(“SECCDED”)错误校正码(“ECC”)以保护数据。在至少一个实施例中,ECC可以为对数据损坏敏感的计算应用程序提供了更高的可靠性。

[0369] 在至少一个实施例中,PPU实现多级存储器层次结构。在至少一个实施例中,内存分区单元4400支持统一存储器以为CPU和PPU存储器提供单个统一虚拟地址空间,从而实现虚拟存储器系统之间的数据共享。在至少一个实施例中,追踪PPU对位于其他处理器上的存储器的访问频率,以确保将存储器页面移动到更频繁地访问页面的PPU的物理存储器。在一个实施例中,高速GPU互连4208支持地址转换服务,该地址转换服务允许PPU直接

访问CPU的页表并提供由PPU对CPU存储器的完全访问。

[0370] 在一个实施例中,复制引擎在多个PPU之间或在PPU与CPU之间 传送数据。在一个实施例中,复制引擎可以为未被映射到页面表中的地址 生成页面错误,并且内存分区单元4400然后为页面错误提供服务,将地址 映射到页面表中,之后复制引擎执行传输。在至少一个实施例中,为多个 处理器之间的多个复制引擎操作固定(即,不可分页)存储器,从而实质上减少了可用存储器。在一个实施例中,由于硬件页面故障,可以将地址 传递给复制引擎,而不必考虑存储页面是否驻留,并且复制过程是透明的。

[0371] 根据至少一个实施例,来自图42的存储器4204或其他系统存储器 的数据由内存分区单元4400提取并存储在L2高速缓存4404中,该L2高速缓存4404位于芯片上并且在各种GPC之间共享。在一个实施例中,每个内存分区单元4400包括与对应的存储器设备相关联的L2高速缓存的至少一部分。在至少一个实施例中,在GPC内的各个单元中实现较低级别的高速缓存。在一个实施例中,每个SM 4314可以实现一级(“L1”)高速缓存,其中L1高速缓存是专用于特定SM 4314的私有存储器,并且获取来自L2高速缓存4404的数据并将其存储在每个L1高速缓存用于在SM 4314 的功能单元中进行处理。在一个实施例中,L2高速缓存4404耦合到存储器接口4406和XBar 4220。

[0372] 在一个实施例中,ROP单元4402执行与像素颜色有关的图形光栅操作,诸如颜色压缩、像素混合等。在一个实施例中,ROP单元4402与光栅引擎4308一起实施深度测试,从光栅引擎4308的剔除引擎接收与像素 片段相关联的样本位置的深度。在至少一个实施例中,为针对与片段相关联的样本位置,在深度缓冲区中针对相应深度进行了深度测试。在至少一个实施例中,如果片段通过了针对样本位置的深度测试,则ROP单元4402 更新深度缓冲器,并将深度测试的结果发送至栅格光栅引擎4308。将理解 的是,分区单元4400的数量可以与GPC的数量不同,因此,在至少一个 实施例中,每个ROP单元4402可以耦合到每个GPC。在至少一个实施例中,ROP单元4402追踪从不同GPC接收到的分组,并确定通过Xbar 4220 将ROP单元4402生成的结果路由到哪个GPC。图45示出了根据一个实施例的流传输多处理器(“SM”)4500。在至少一个实施例中,SM 4500 是图43的SM 4314。在至少一个实施例中,SM 4500包括但不限于,指令 缓存4502;一个或更多个调度器单元4504;寄存器文件4508;一个或更多个处理核心(“核心”)4510;一个或更多个特殊功能单元(“SFU”) 4512;一个或更多个加载/存储单元(“LSU”) 4514;互连网络4516;共享存储器/一级(“L1”)高速缓存4518;和任何它们的适当组合。

[0373] 在至少一个实施例中,工作分配单元分派任务以在并行处理单元(“PPU”)的通用处理集群(“GPC”)上执行,并且每个任务被分配 给GPC内的特定数据处理集群(“DPC”),以及,如果该任务与着色器 程序相关联,则该任务被分配给SM 4500。在一个实施例中,调度器单元 4504从工作分配单元接收任务,并管理分配给SM 4500的一个或更多个线程块的指令调度。在至少一个实施例中,调度器单元4504调度线程块以作为并行线程的线程束进行执行,其中每个线程块至少分配一个线程束。在至少一个实施例中,每个线程束执行线程。在至少一个实施例中,调度器 单元4504管理多个不同的线程块,将线程束分配给不同的线程块,然后在 每个时钟周期将来自多个不同的协作组的指令分配给各个功能单元(例如,核心4510、SFU 4512和LSU 4514)。

[0374] 在至少一个实施例中,协作组可以指用于组织通信线程的组的编程 模型,该编程

模型允许开发人员表达正在通信的线程的粒度,从而使表达更丰富,更有效的并行分解。在至少一个实施例中,协作启动API支持线程块之间的同步以执行并行算法。在至少一个实施例中,常规编程模型的应用程序提供了用于同步协作线程的单个简单构造:跨线程块的所有线程的屏障(例如, `syncthreads()` 函数)。在至少一个实施例中,编程人员经常想以小于线程块粒度的大小来定义线程组,并在定义的组内进行同步,从而以集体范围内的功能接口的形式实现更高的性能、设计灵活性和软件重用。在至少一个实施例中,协作组使编程人员定义明确位于子块(例如,小到单个线程)和多块粒度上的线程组,并在协作组中对线程执行集体操作,例如同步。在至少一个实施例中,编程模型支持跨软件边界的清除合成,因此库和实用功能可以在其本地上下文中安全地同步,而不必对收敛进行假设。在至少一个实施例中,协作组基元启用了新的合作并行模式,包括生产者-消费者并行,机会主义并行以及整个线程块网格上的全局同步。

[0375] 在至少一个实施例中,调度单元4506配置成将指令发送到一个或多个功能单元,并且调度器单元4504包括但不限于两个调度单元4506,该两个调度单元4506使得来自同一线程束的两个不同指令能够在每个时钟周期内被调度。在至少一个实施例中,每个调度器单元4504包括单个调度单元4506或附加调度单元4506。

[0376] 在至少一个实施例中,每个SM 4500包括寄存器文件4508,其为SM 4500的功能单元提供一组寄存器。在至少一个实施例中,寄存器文件4508在每个功能单元之间划分,使得每个功能单元被分配寄存器文件4508的专用部分。在至少一个实施例中,寄存器文件4508被SM 4500执行的不同线程束划分,以及寄存器文件4508为连接到功能单元的数据路径的操作数提供临时存储。在至少一个实施例中,每个SM 4500包括但不限于多个数量为L的处理核心4510。在至少一个实施例中,SM 4500包括但不限于大量(例如,128个或更多个)不同的处理核心4510。在至少一个实施例中,每个核心4510包括但不限于全管线、单精度、双精度和/或混合精度处理单元,处理单元包括但不限于浮点运算逻辑单元(“ALU”)和整数算术逻辑单元。在至少一个实施例中,浮点算术逻辑单元实施IEEE 754-2008标准用于浮点算术。在至少一个实施例中,处理核心4510包括但不限于64个单精度(32位)浮点核心、64个整数核心、32个双精度(64位)浮点核心和8个张量核心。

[0377] 根据至少一个实施例,张量核心配置成为执行矩阵运算。在至少一个实施例中,核心4510中包括一个或多个张量核心。在至少一个实施例中,张量核心配置成执行深度学习矩阵算术,例如用于神经网络训练和推理的卷积运算。在至少一个实施例中,每个张量核心在 4×4 矩阵上操作并执行矩阵乘法和累加运算 $D=A \times B+C$,其中A,B,C和D是 4×4 矩阵。

[0378] 在至少一个实施例中,矩阵乘法输入A和B是16位浮点矩阵,并且累加矩阵C和D是16位浮点或32位浮点矩阵。在至少一个实施例中,张量核心对16位浮点输入数据进行32位浮点累加运算。在至少一个实施例中,16位浮点乘法需要64次运算,并产生全精度乘积,全精度乘积然后使用32位浮点加法与其他用于 $4 \times 4 \times 4$ 矩阵的中间乘积累加。在至少一个实施例中,张量核心用于执行由这些较小的元件构成的更大的二维或更高维度的矩阵运算。在至少一个实施例中,诸如CUDA 9C++API之类的API公开专门的矩阵加载、矩阵乘法和累加以及矩阵存储操作,以有效地使用来自ClJDA-C++程序的张量核心。在至少一个实施例中,在CUDA级别,线程束级别接口假设 16×16 大小的矩阵跨线程束的所有32个线程。

[0379] 在至少一个实施例中,每个SM 4500包括但不限于执行特殊功能(例如,属性评

估、倒数平方根等)的M个SFU 4512。在至少一个实施例中, SFU 4512包括但不限于配置成传递分层树数据结构的树传递单元。在至少一个实施例中, SFU 4512包括但不限于配置成执行纹理图过滤操作的纹理单元。在至少一个实施例中, 纹理单元配置成从存储器加载纹理图(例如, 像素的2D阵列)并且对纹理图进行采样以产生采样的纹理值以供在由SM 4500执行的着色器程序中使用。在至少一个实施例中, 纹理图存储在共享存储器/L1高速缓存4518中。在至少一个实施例中, 纹理单元实施纹理操作, 例如使用mip-maps(例如, 细节水平变化的纹理图)进行滤波操作。在至少一个实施例中, 每个SM 4500包括但不限于两个纹理单元。

[0380] 在至少一个实施例中, 每个SM 4500包括但不限于N个LSU 4514, 这些LSU 4514实现共享存储器/L1高速缓存4518和寄存器文件4508之间的加载和存储操作。在至少一个实施例中, 每个SM 4500包括但不限于互连网络4516, 互连网络4516将每个功能单元连接到寄存器文件4508和将LSU 4514连接到寄存器文件4508和共享存储器/L1高速缓存4518。在至少一个实施例中, 互连网络4516是可配置为将任何功能单元连接到寄存器堆4508中的任何寄存器并将LSU 4514连接到寄存器文件4508和共享存储器/L1高速缓存4518中的存储器位置的交叉开关。

[0381] 在至少一个实施例中, 共享存储器/L1高速缓存4518是片上存储器的阵列, 其在一个实施例中允许SM 4500与原始引擎之间以及SM 4500中的线程之间的数据存储和通信。在至少一个实施例中, 共享存储器/L1高速缓存4518包括但不限于128KB的存储容量, 并且位于从SM 4500到分区单元的路径中。在至少一个实施例中, 共享存储器/L1高速缓存4518用于高速缓存读取和写入。共享存储器/L1高速缓存4518、L2高速缓存和存储器中的一个或更多个是后备存储器。

[0382] 在至少一个实施例中, 将数据高速缓存和共享存储器功能组合到单个存储器块中, 为两种类型的存储器访问提供了改进的性能。在至少一个实施例中, 该容量被不使用共享存储器的程序使用或用作高速缓存, 例如, 如果将共享存储器配置为使用一半容量, 则纹理和加载/存储操作可以使用剩余容量。根据至少一个实施例, 在共享存储器/L1高速缓存4518内的集成使共享存储器/L1高速缓存4518能够用作用于流传输数据的高吞吐量管线, 同时提供对频繁重用的数据的高带宽和低延迟访问。在至少一个实施例中, 当配置用于通用并行计算时, 与图形处理相比, 可以使用更简单的配置。在至少一个实施例中, 固定功能图形处理单元被绕过, 从而创建了更简单的编程模型。在至少一个实施例中, 在通用并行计算配置中, 工作分配单元直接将线程的块分配和分布给DPC。在至少一个实施例中, 块中的线程执行相同的程序, 在计算中使用唯一的线程ID来确保每个线程生成唯一的结果, 使用SM 4500执行程序并执行计算, 使用共享存储器/L1高速缓存4518在线程之间进行通信, 以及LSU 4514通过共享存储器/L1高速缓存4518和内存分区单元来读写全局存储器。在至少一个实施例中, 当被配置用于通用并行计算时, SM 4500写入调度器单元4504可以用来在DPC上启动新工作的命令。

[0383] 在至少一个实施例中, PPU被包括在或耦合到台式计算机、膝上型计算机、平板电脑、服务器、超级计算机、智能电话(例如, 无线、手持设备)、个人数字助理(“PDA”)、数码相机、车辆、头戴式显示器、手持式电子设备等中或与之耦合。在至少一个实施例中, PPU被实现在单个半导体衬底上。在至少一个实施例中, PPU与一个或更多个其他设备(例如附加

的PPU、存储器、精简指令集计算机(“RISC”)CPU,一个或更多个存储器管理单元(“MMU”)、数模转换器(“DAC”)等)一起被包括在片上系统(“SoC”)中。

[0384] 在至少一个实施例中,PPU可以被包括在包括一个或更多个存储设备的图形卡上。图形卡可以配置为与台式计算机主板上的PCIe插槽相连接。在至少一个实施例中,PPU可以是包括在主板的芯片组中的集成图形处理单元(“iGPU”)。

[0385] 推理和/或训练逻辑1015用于执行与一个或更多个实施例相关的推理和/或训练操作。本文结合图9和/或图10提供关于推理和/或训练逻辑1015的细节。在至少一个实施例中,推理和/或训练逻辑1015可以用于推理或预测软件程序以使用优化的决策树解决问题。在至少一个实施例中,推理和/或训练逻辑1015可以用于选择用于解决特定操作的内核。

[0386] 在至少一个实施例中,单个半导体平台可以指单个统一的基于半导体的集成电路或芯片。在至少一个实施例中,可以使用具有增强的连接性的多芯片模块,其模拟芯片上的操作,并且相对于利用常规的中央处理单元(“CPU”)和总线实现方式进行了实质性的改进。在至少一个实施例中,根据用户的需求,各种模块也可以分开放置或以半导体平台的各种组合放置。

[0387] 在至少一个实施例中,以机器可读可执行代码或计算机控制逻辑算法形式的计算机程序被存储在主存储器1404和/或辅助存储器中。根据至少一个实施例,如果由一个或更多个处理器执行,则计算机程序使系统1400能够执行各种功能。在至少一个实施例中,主存储器1404、存储和/或任何其他存储器是计算机可读介质的可能示例。在至少一个实施例中,辅助存储器可以指任何合适的存储设备或系统,例如硬盘驱动器和/或可移除存储驱动器,代表软盘驱动器、磁带驱动器、光盘驱动器、数字通用盘(“DVD”)驱动器、记录设备、通用串行总线(“USB”)闪存等。在至少一个实施例中,在以下项的上下文中实现了各种先前附图的架构和/或功能:CPU 1402;并行处理系统1412;具有两个CPU 1402的至少一部分能力的集成电路;并行处理系统1412;芯片组(例如,一组被设计为用于执行相关功能的单元工作并出售的集成电路等);以及集成电路的任何适当组合。

[0388] 在至少一个实施例中,各种先前附图的架构和/或功能是在通用计算机系统、电路板系统、专用于娱乐目的的游戏控制台系统、专用系统等背景下实现的。在至少一个实施例中,计算机系统1400可以采取台式机、膝上型计算机、平板计算机、服务器、超级计算机、智能电话(例如,无线、手持设备)、个人数字助理(“PDA”)、数码相机、车辆、头戴式显示器、手持式电子设备、移动电话设备、电视、工作站、游戏机、嵌入式系统和/或任何其他类型的逻辑的形式。

[0389] 在至少一个实施例中,并行处理系统1412包括但不限于多个并行处理单元(“PPU”)1414和相关联的存储器1416。在至少一个实施例中,PPU 1414通过互连1418和开关1420或多路复用器连接到主机处理器或其他外围设备。在至少一个实施例中,并行处理系统1412在可并行化的PPU 1414上分布计算任务,例如,作为跨多个图形处理单元(“GPU”)线程块的计算任务分布的一部分。在至少一个实施例中,存储器在一些或全部PPU 1414之间是共享和可访问的(例如,用于读取和/或写入访问),尽管这样的共享存储器可能引发相对于使用本地存储器和驻留在PPU1414上的寄存器的性能损失。在至少一个实施例中,通过使用诸如__syncthreads()之类的命令来同步PPU 1414的操作,其中块中的所有

线程(例如,跨多个PPU 1414执行)在进行之前到达代码的某个执行点。

[0390] 虚拟计算平台

[0391] 公开了实施例,其涉及用于高级计算的虚拟化计算平台,诸如医学 应用程序中的图像推理和图像处理。没有限制,实施例可以包括放射线照 相、磁共振成像(MRI)、核医学、超声、超声检查、弹性成像、光声成 像、断层扫描、超声心动图、功能近红外光谱和磁性粒子成像,或其组合。在至少一个实施例中,本文所述的虚拟化计算平台和相关过程可以附加地 或替代地用于但不限于法医科学分析、地下探测和成像(例如,石油勘探、考古、古生物学等)、地形、海洋学、地质学、骨病学、气象学、智能区 域或对象跟踪和监视、传感器数据处理(例如RADAR、SONAR、LIDAR 等)和/或基因组学和基因测序。

[0392] 参见图46,图46是根据至少一个实施例的用于生成和部署图像处理 和推理管线的过程4600的示例数据流程图。在至少一个实施例中,过程 4600可以被部署用于在一个或更多个设施4602(例如,医疗设施、医院、 医疗机构、诊所、研究或诊断实验室等)处的成像设备、处理设备、基因 设备、基因序列设备、放射线设备和/或其他设备等。在至少一个实施 例中,过程4600可以被部署以对测序数据进行基因组分析和推理。可以使用本文 所述的系统和过程进行的基因组分析的实例包括但不限于变体调用、突变 检测和基因表达定量。

[0393] 过程4600可以在训练系统4604和/或部署系统4606内执行。在至少 一个实施例中,训练系统4604可以用于执行在部署系统4606中使用的机 器学习模型的训练,部署和实现(例如,神经网络、对象检测算法、计算 机视觉算法等)。在至少一个实施例中,部署系统 4606可配置为在分布式 计算环境中卸载处理和计算资源以减少设施4602上的基础设施需求。在至 少一个实施例中,部署系统4606可以提供用于选择、定制和实现与成像设 备(例如,MRI、CT扫描、X-射线、超声等)一起使用的虚拟仪器或在 设施4602处对设备进行排序的简化平台。在至少一个实施例中,虚拟仪器 可以包括软件定义的应用程序,用于执行关于 由成像设备、测序设备、放 射线设备和/或其他设备类型生成的成像数据的分类的一个或更多个处理 操作的。在至少一个实施例中,管线中的一个或更多个应用可以在应用的 执行期间使用或调用部署系统4606的服务(例如,推理、可视化、计算、AI等)。

[0394] 在至少一个实施例中,在高级处理和推理管线中使用的一些应用程 序可以使用机器学习模型或其他AI来执行一个或更多个处理步骤。在至少 一个实施例中,可以使用在设施4602处生成(并且存储在设施4602处的 一个或更多个图片归档和通信系统(PACS)服务器上)的数据4608(例 如成像数据)在设施4602处训练机器学习模型。使用来自另一个设 施或多个设施(例如,不同的医院、实验室、诊所等)或其组合的成像或测序数 据4608进行训练。在至少一个实施例中,训练系统4604可以用于提供应 用程序、服务和/或其他资源,以生成用于部署系统4606的工作的、可部 署的机器学习模型。

[0395] 在至少一个实施例中,模型注册表4624可以由对象存储支持,该对 象存储可以支持版本控制和对象元数据。在至少一个实施例中,可以通过 例如来自云平台内的云存储(例如,图47的云4726)兼容的应用程序编 程接口(API)来访问对象存储。在至少一个实施 例中,模型注册表4624 中的机器学习模型可以由与API交互的系统的开发者或合作伙伴上 载、列 出、修改或删除。在至少一个实施例中,API可以提供对方法的访问,该 方法允许具有适当证书的用户将模型与应用程序相关联,使得可以将模型 作为应用程序的容器化实 例的执行的一部分来执行

[0396] 在至少一个实施例中,训练管线4704(图47)可以包括以下情形:设施4602正在训练他们自己的机器学习模型,或者具有需要优化或更新的 现有机器学习模型。在至少一个实施例中,可以接收由成像设备、测序设 备和/或其他设备类型生成的成像数据4608。在至少一个实施例中,一旦接 收到成像数据4608,AI辅助注释4610就可以用于帮助生成与成像数据 4608相对应的注释,以用作机器学习模型的地面实况数据。在至少一个实 施例中,AI辅助注释4610可以包括一个或更多个机器学习模型(例如, 卷积神经网络(CNN)),可以对机器学习模型进行训练以生成对应于某 些类型的成像数据4608(例如,来自某些设备)和/或成像数据4608中某 些类型的异常的注释。在至少一个实施例中,然后可以直接使用AI辅助注 释4610,或者可以使用注释工具(例如,由研究人员、临床医生、医生、科学家等)调整或精调,以生成地面实况数据。在至少一个实施例中,在 一些示例中,标记的临床数据4612(例如,由临床医生、医生、科学家、技术人员等提供的注释)可以用作用于训练机器学习模型的地面实况数据。在至少一个实施例中,AI辅助注释4610、标记的临床数据4612或其组合 可以用作训练机器学习模型的地面实况数据。在至少一个实施例中,训练 的机器学习模型可以被称为输出模型4616,并且可以由部署系统4606使 用,如本文所述。

[0397] 在至少一个实施例中,训练管线4704(图47)可以包括以下场景:设施4602需要机器学习模型以用于为部署系统4606中的一个或更多个应 用程序执行一个或更多个处理任务,但是设施4602当前可能没有这样的机 器学习模型(或者可能没有为此目的而优化、高效或有效的模型)。在至 少一个实施例中,可以从模型注册表4624中选择现有的机器学习 模型。在 至少一个实施例中,模型注册表4624可以包括经训练以对成像数据执行各 种不同的推理任务的机器学习模型。在至少一个实施例中,可以在来自不 同于设施4602的设施(例如,位于远程的设施)的成像数据上训练模型注 册表4624中的机器学习模型。在至少一个实施例中,机器学习模型可能已 经在来自一个位置、两个位置或任何数量的位置的成像数据上进行训练。在至少一个实施例中,当从特定位置对成像数据进行训练时,可以在该位 置进行训练,或者至少以保护成像数据的机密性或限制来自备用设备(off-premises)转移的成像数据的方式进行训练(例如,遵守HIPAA法规、隐私法规等)。在至少一个实施例中,一旦在一个位置上训练了模型或部 分地训练了模型,就可以将机器学习模型添加到模型注册表4624中。在至 少一个实施例中,然后可以在任何数量的其他设施上对机器学习模型进行 重新训练或更新,可以在模型注册表4624中使用经过重新训练或更新的模 型。在至少一个实施例中,然后可以从模型注册表4624中选择机器学习模 型,并称为输出模型4616,并且可以在部署系统4606中使用机器学习模 型来为部署系统的一个或更多个应用程序执行一个或更多个处理任务。在 至少一个实施例中,可以在包括设施4602的场景中使用训练管线4704(图 47),该设施需要机器学习模型以用于对部署系统4606中的一个或更 多个 应用程序执行一个或更多个处理任务,但是设施4602当前可能没有这样的 机器学习模型(或者可能没有为此目的而优化、高效或有效的模型)。在 至少一个实施例中,由于群体差异、基因变异,用于训练机器学习模型的 训练数据的鲁棒性、训练数据异常的多样性和/或训练数据的其他问题,从 模型注册表4624中选择的机器学习模型可能不会针对设施 4602处生成的 成像数据4608进行微调或优化。在至少一个实施例中,AI辅助注释4610 可以用于帮助生成与成像数据4608相对应的注释,以将其用作地面实况数 据以用于重新训练或更新机器学习模型。在至少一个实施例中,标记的临 床数据4612(例如,由临床医生、

医生、科学家等提供的注释)可以用作训练机器学习模型的地面实况数据。在至少一个实施例中,重新训练或更新机器学习模型可以称为模型训练4614。在至少一个实施例中,模型训练4614-例如,AI辅助注释4610、标记的临床数据4612或其组合-可以用作训练或更新机器学习模型的地面实况数据。在至少一个实施例中,训练的机器学习模型可以被称为输出模型4616,并且可以由部署系统4606使用,如本文所述。

[0398] 在至少一个实施例中,部署系统4606可以包括软件4618、服务4620、硬件4622和/或其他组件、特征和功能。在至少一个实施例中,部署系统4606可以包括软件“堆栈”,使得软件4618可以建立在服务4620的顶部并且可以使用服务4620来执行一些或全部处理任务,并且服务4620和软件4618可以构建在硬件4622之上,并使用硬件4622执行部署系统4606的处理、存储和/或其他计算任务。在至少一个实施例中,软件4618可以包括任何数量的不同容器,其中每个容器可以执行应用程序的实例化。在至少一个实施例中,每个应用程序可以在高级处理和推理管线中执行一个或多个处理任务(例如,推理、对象检测、特征检测、分割、图像增强、校准等)。在至少一个实施例中,对于每种类型的成像设备(例如,CT、MRI、X射线、超声、超声检查、超声心动图等)、测序设备、放射学设备、基因组学设备等,可以存在可以针对由设备生成的成像数据4608(或其他数据类型,例如本文所述的那些数据)执行数据处理任务的任何数量的容器。在至少一个实施例中,除了接收和配置成像图像以供每个容器使用和/或设施4602在通过管线处理后使用(例如,将输出转换回可用的数据类型,例如医学数字成像和通信(DICOM)数据、放射信息系统(RIS)数据、临床信息系统(CIS)数据、远程过程调用(RPC)数据、基本符合表示状态传输(REST)接口的数据、基本符合基于文件的接口的数据和/或原始数据,以便在设施4602进行存储和显示)的容器之外,还可以基于对处理成像数据4608期望或要求的不同容器的选择来定义高级处理和推理管线。在至少一个实施例中,软件4618(例如,组成管线)内的容器的组合可以被称为虚拟仪器(如本文中更详细描述),并且虚拟仪器可以利用服务4620和硬件4622执行在容器中实例化的应用程序的部分或全部处理任务。

[0399] 在至少一个实施例中,数据处理管线可以响应于推理请求(例如,来自部署系统4606的用户的请求,例如临床医生、医生、放射线医生等)而以DICOM、RIS、CIS、REST兼容、RPC、原始和/或其他格式接收输入数据(例如,成像数据4608)。在至少一个实施例中,输入数据可以代表由一个或多个成像设备、测序设备、放射学设备、基因组设备和/或其他设备类型生成的一个或多个图像、视频和/或其他数据表示。在至少一个实施例中,数据可以作为数据处理管线的一部分进行预处理,以准备数据以供一个或多个应用程序处理。在至少一个实施例中,可以对管线的一个或多个推理任务或其他处理任务的输出执行后处理,以为下一应用程序准备输出数据和/或准备输出数据用于传送和/或由用户使用(例如,作为推理请求的响应)。在至少一个实施例中,推理任务可以由一个或多个机器学习模型来执行,例如训练或部署的神经网络,其可以包括训练系统4604的输出模型4616。

[0400] 在至少一个实施例中,数据处理管线的任务可以封装在一个或多个容器中,每个容器表示能够引用机器学习模型的应用程序和虚拟化计算环境的离散的、全功能的实例化。在至少一个实施例中,容器或应用程序可以被发布到容器注册表的私有(例如,有限访问)区域中(在此更详细地描述),并且训练或部署的模型可以被存储在模型注册表4624

中并且与一个或多个应用程序相关联。在至少一个实施例中,应用程序的图像(例如,容器图像)可以在容器注册表中可用,并且一旦被用户从容器注册表中选择以部署在管线中,则可以使用图像来生成容器以用于对由用户系统使用的应用程序进行实例化。

[0401] 在至少一个实施例中,开发人员(例如,软件开发人员、临床医师、医生等)可以开发、发布和存储用于对所提供的数据执行图像处理 and/或推理的应用程序(例如,作为容器)。在至少一个实施例中,可以使用与系统相关联的软件开发套件(SDK)来执行开发、发布和/或存储(例如,以确保开发的应用程序和/或容器与系统兼容或相匹配)。在至少一个实施例中,可以使用可以支持至少一些服务4620作为系统(例如,图47的系统4700)的SDK在本地(例如,在第一设施处,在来自第一设施的数据上)测试开发的应用程序。在至少一个实施例中,因为DICOM对象可以包含一到数百个图像或其他数据类型的任何地方,并且由于数据的变化,所以开发人员可以负责管理(例如,设置构造,将预处理构建为应用程序等)提取和准备传入的DICOM数据。在至少一个实施例中,一旦通过系统4700的验证(例如,准确性、安全性、患者隐私等),应用程序就可以在容器注册表中用于用户(例如,医院、诊所、实验室、医疗保健提供者等)以针对用户设施(例如第二设施)上的数据执行一个或多个处理任务。

[0402] 在至少一个实施例中,开发人员然后可以通过网络共享应用程序或容器,以供系统(例如,图47的系统4700)的用户访问和使用。在至少一个实施例中,可以将完成并经过验证的应用程序或容器存储在容器注册表中,并且可以将相关的机器学习模型存储在模型注册表4624中。在至少一个实施例中,请求实体(例如,医疗机构中的用户)(提供推理或图像处理请求的人)可以浏览容器注册表和/或模型注册表4624以查找应用程序、容器、数据集、机器学习模型等,选择所需的元素组合以包含在数据处理管线中,并提交成像处理请求。在至少一个实施例中,请求可以包括执行请求所必需的输入数据(以及在一些示例中,与患者相关的数据),和/或可以包括选择对应用程序和/或机器学习模型以在处理请求时执行。在至少一个实施例中,然后将请求传递到部署系统4606的一个或多个组件(例如,云)以执行数据处理管线的处理。在至少一个实施例中,由部署系统4606进行的处理可以包括从容器注册表和/或模型注册表4624中引用所选择的元素(例如,应用、容器、模型等)。在至少一个实施例中,一旦由在管线生成结果,结果可以返回给用户以供参考(例如,用于在本地,本地部署(on-premises)工作站或终端上执行的查看应用程序套件中进行查看)。在至少一个实施例中,放射线医师可以从包括任何数量的应用程序和/或容器的数据处理管线接收结果,其中结果可以包括X射线、CT扫描、MRI等的异常检测。

[0403] 在至少一个实施例中,为了帮助处理或执行管线中的应用程序或容器,可以利用服务4620。在至少一个实施例中,服务4620可以包括计算服务、人工智能(AI)服务、可视化服务和/或其他服务类型。在至少一个实施例中,服务4620可以提供软件4618中的一个或多个应用所共有的功能,因此可以将功能抽象为可以被应用程序调用或利用的服务。在至少一个实施例中,由服务4620提供的功能可以动态且更有效地运行,同时还通过允许应用程序并行处理数据(例如,使用并行计算平台4730(图47))来很好地缩放。在至少一个实施例中,不是要求共享服务4620所提供的相同功能的每个应用程序具有服务4620的相应实例,而是可以在各种应用程序之间共享服务4620。在至少一个实施例中,作为非限制性示例,服务可以包括可以用于执行检测或分割任务的推理服务器或引擎。在至少一个实

施例中,可以包括模型训练服务,其可以提供机器学习模型训练和/或重新训练能力。在至少一个实施例中,可以进一步包括数据增强服务,其可以提供GPU加速的数据(例如,DICOM、RIS、CIS、REST兼容、RPC、原始等),提取、调整大小、缩放和/或其他增强。在至少一个实施例中,可以使用可视化服务,该可视化服务可以添加图像渲染效果(例如,射线追踪、光栅化、去噪、锐化等),以向二维(2D)和/或三维(3D)模型添加真实感。在至少一个实施例中,可以包括虚拟仪器服务,其为虚拟仪器的管线内的其他应用程序提供波束形成、分段、推理、成像和/或支持。

[0404] 在至少一个实施例中,在服务4620包括AI服务(例如,推理服务)的情况下,与用于异常检测(例如,肿瘤、生长异常、瘢痕形成等)的应用程序相关联的一个或多个机器学习模型可以通过调用(例如,作为API调用)推理服务(例如,推理服务器)以执行机器学习模型或其处理,作为应用程序执行的一部分来执行。在至少一个实施例中,在另一应用程序包括用于分割任务的一个或多个机器学习模型的情况下,应用程序可以调用推理服务来执行用于执行与分割任务相关联的一个或多个处理操作的机器学习模型。在至少一个实施例中,可以使实现包括分段应用程序和异常检测应用程序的高级处理和推理管线的软件4618成流线型,因为每个应用程序可以调用相同的推理服务来执行一个或多个推理任务。在至少一个实施例中,硬件4622可以包括GPU、CPU、图形卡、AI/深度学习系统(例如,AI超级计算机、例如NVIDIA的DGX超级计算机系统)、云平台或其组合。在至少一个实施例中,可以使用不同类型的硬件4622为部署系统4606中的软件4618和服务4620提供有效的、特定目的的支持。在至少一个实施例中,可以实现GPU处理的使用以进行本地处理(例如在设施4602),在AI/深度学习系统中、在云系统中和/或在部署系统4606的其他处理组件中,以提高图像处理、图像重建、分割、MRI检查、中风或心脏病发作检测(例如实时)、渲染中的图像质量等的效率、准确性和功效。在至少一个实施例中,设施可以包括成像设备、基因组设备、测序设备和/或可以利用GPU生成代表受试者解剖结构的成像数据其他本地部署设备类型。

[0405] 在至少一个实施例中,作为非限制性示例,可以针对深度学习、机器学习和/或高性能计算来优化软件4618和/或服务4620以用于GPU处理。在至少一个实施例中,部署系统4606和/或训练系统4604的计算环境中的至少一些可以在数据中心中执行具有GPU优化的软件(例如,NVIDIA的DGX系统的硬件和软件的组合)的一个或多个超级计算机或高性能计算系统。在至少一个实施例中,数据中心可以符合HIPAA的规定,使得针对患者数据的隐私性安全地处理接收、处理和传输成像数据和/或其他患者数据。在至少一个实施例中,硬件4622可包括可被调用以并行执行数据处理的任意数量的GPU,如本文所述。在至少一个实施例中,云平台还可包括用于深度学习任务、机器学习任务或其他计算任务的GPU优化执行的GPU处理。在至少一个实施例中,云平台(例如,NVIDIA的NGC)可以使用作为硬件抽象和缩放平台AI/深度学习超级计算机和/或GPU优化的软件(例如,如在NVIDIA的DGX系统上提供的)来执行。在至少一个实施例中,云平台可以在多个GPU上集成应用程序容器集群系统或编排系统(例如,KUBERNETES),以实现无缝缩放和负载平衡。

[0406] 图47是根据至少一个实施例的用于生成和部署成像部署管线的示例系统4700的系统图。在至少一个实施例中,系统4700可以用于实现图46的过程4600和/或其他过程,包括高级处理和推理管线。在至少一个实施例中,系统4700可以包括训练系统4604和部署系

统4606。在至少一个实施例中,可以使用软件4618、服务4620和/或硬件4622来实现训练系统 4604和部署系统4606。

[0407] 在至少一个实施例中,系统4700(例如,训练系统4604和/或部署 系统4606)可以在云计算环境中(例如,使用云4726)实现。在至少一个 实施例中,系统4700可以针对医疗服务设施在本地实现,或者作为云和本 地计算资源的组合来实现。在至少一个实施例中,在实施云计算实现的实 施例中,患者数据可以与系统4700的一个或更多个组件分离或不被系统 4700的一个或更多个组件处理,这将使得处理不符合HIPAA和/或其他数 据处理和隐私法规或法律。在至少一个实施例中,可以通过制定的安全措 施或协议将对云4726中的API的访问限制为授权用户。在至少一个实施 例中,安全协议可以包括可以由认证服务(例如,AuthN,AuthZ,Gluecon 等)签名的网络令牌,并且可以携带适当的授权。在至少一个实 施例中,虚拟仪器的API(在此描述)或系统4700的其他实例可以被限制到已经被 审核或被授权用于交互的一组公共IP。

[0408] 在至少一个实施例中,系统4700的各个组件可以使用各种不同的网 络类型(包括但不限于局域网(LAN)和/或广域网(WAN))中的任何 一种经由线和/或无线通信协议在彼此之间进行通信。在至少一个实施例中, 系统4700的设施和组件之间的通信(例如,用于发送推理请求,用于接收 推理请求的结果等)可以通过一条或多条数据总线、无线数据协议 (Wi-Fi)、有线数据协议(例如以太网)等进行通信。

[0409] 在至少一个实施例中,训练系统4604可以执行训练管线4704,类似 于本文中相对于图46所描述的那些。在至少一个实施例中,其中部署系统 4606将在部署管线4710中使用一个或更多个机器学习模型,训练管线4704 可用于训练或重新训练一个或更多个(例如,预训练的)模型,和/或实现 一个或多个预训练模型4706(例如,无需重新训练或更新)。在 至少一个 实施例中,作为训练管线4704的结果,可以生成输出模型4616。在至少 一个实施 例中,训练管线4704可以包括任何数量的处理步骤,诸如但不限 于成像数据(或其他输入 数据)的转换或适应(例如,使用DICOM适配 器4702A将DICOM图像转换为适合于通过相应的 机器学习模型进行处理, 例如神经影像信息技术计划(NIfTI)格式,AI辅助注释4610、成像 数据 4608的标签或注释以生成标记的临床数据4612,从模型注册表中选择模型, 模型训 练4614,训练、重新训练或更新模型和/或其他处理步骤。在至少一 个实施例中,对于由部 署系统4606使用的不同的机器学习模型,可以使用 不同的训练管线4704。在至少一个实施 例中,类似于关于图46描述的第一示例的训练管线4704可以被用于第一机器学习模型,类 似于关于图46 描述的第二示例的训练管线4704可以被用于第二机器学习模型,类似于关 于图46描述的第三示例的训练管线4704可以被用于第三机器学习模型。在至少一个实施 例中,可以根据每个相应的机器学习模型的要求来使用训 练系统4604内的任务的任何组 合。在至少一个实施例中,机器学习模型中 的一个或更多个可以已经被训练并且准备好部 署,因此机器学习模型可以 不由训练系统4604进行任何处理,并且可以由部署系统4606来 实现。

[0410] 在至少一个实施例中,取决于实施方式或实施例,输出模型4616和 /或预训练模 型4706可以包括任何类型的机器学习模型。在至少一个实施 例中,但不限于,系统4700使 用的机器学习模型可以包括使用线性回归、逻辑回归、决策树、支持向量机(SVM)、朴素贝 叶斯、k最近邻(Knn)、K表示聚类、随机森林、降维算法、梯度提升算法、神经网络(例如,自

动编码器、卷积、递归、感知器、长期/短期记忆 (LSTM)、Hopfield、Boltzmann、深层信念、反卷积、生成对抗、液体状态机等) 和/或其他类型的机器学习模型。

[0411] 在至少一个实施例中,训练管线4704可以包括AI辅助注释,如本文相对于至少图50B更详细地描述的。在至少一个实施例中,可以通过许多技术来生成标记的临床数据4612(例如,传统注释)。在至少一个实施例中,可以在绘图程序(例如,注释程序)、计算机辅助设计(CAD)程序、标记程序、适合于为地面实况生成注释或标签的另一种类型的程序内生成标签或其他注释,和/或在一些示例中,标签或其他注释可以是手绘的。在至少一个实施例中,地面实况数据可以是合成产生的(例如,从计算机模型或渲染产生的)、真实产生的(例如,从真实世界的真实数据设计和产生的)、机器自动化的(例如,使用特征分析和学习以从数据中提取特征,然后生成标签)、人工注释(例如,贴标机或注释专家,定义标签的位置)和/或其组合。在至少一个实施例中,对于成像数据4608(或机器学习模型使用的其他数据类型)的每个实例,可以存在由训练系统4604生成的相应的地面实况数据。在至少一个实施例中,可以使用AI辅助注释作为部署管线4710的一部分执行;或者补充或代替训练管线4704中包括的AI辅助注释。在至少一个实施例中,系统4700可以包括多层平台,该多层平台可以包括执行一项或更多项医学成像和诊断功能的诊断应用程序(或其他应用程序类型)的软件层(例如,软件4618)。在至少一个实施例中,系统4700可以(例如,经由加密的链路)通信地耦合到一个或多个设施的PACS服务器网络。在至少一个实施例中,系统4700可以被配置为从PACS服务器(例如,DICOM数据、RIS数据、原始数据、CIS数据、符合REST的数据、RPC数据、原始数据等)访问和引用数据(例如,通过DICOM适配器4702或其他数据类型(例如RIS、CIS、REST兼容、RPC,原始等)适配器)执行操作,例如训练机器学习模型、部署机器学习模型、图像处理、推理和/或其他操作。

[0412] 在至少一个实施例中,软件层可以被实现为安全、加密和/或认证的API,通过该API可以从外部环境(例如,设施4602)唤醒(例如,调用)应用程序或容器。在至少一个实施例中,应用程序然后可以调用或执行一个或多个服务4620以执行与各个应用程序相关联的计算、AI或可视化任务,并且软件4618和/或服务4620可以利用硬件4622来以有效和高效的方式执行处理任务。

[0413] 在至少一个实施例中,部署系统4606可以执行部署管线4710。在至少一个实施例中,部署管线4710可以包括可以顺序地、非顺序地或以其他方式应用于由成像设备,测序设备,基因组设备等生成的成像数据(包括如上所述的AI辅助注释)的任何数量的应用程序(和/或其他数据类型)。在至少一个实施例中,如本文所述,用于单个设备的部署管线4710可以被称为用于设备的虚拟仪器(例如,虚拟超声仪器、虚拟CT扫描仪器、虚拟测序仪器等)。在至少一个实施例中,对于单个设备,取决于从设备生成数据期望的信息,可以有一个以上的部署管线4710。在至少一个实施例中,在期望从MRI机器检测到异常的情况下,可以存在第一部署管线4710,并且在期望从MRI机器的输出进行图像增强的地方,可以存在第二部署管线4710。

[0414] 在至少一个实施例中,可用于部署管线4710的应用程序可包括可用于对来自设备的成像数据或其他数据执行处理任务的任何应用程序。在至少一个实施例中,不同的应用程序可以负责图像增强、分割、重建、异常检测、物体检测、特征检测、治疗计划、剂量测定、射束计划(或其他放射治疗程序)和/或其他分析、图像处理或推理任务。在至少一个实

施例中，部署系统4606可以为每个应用程序定义构造，使得部署系统4606的用户(例如，医疗设施、实验室、诊所等)可以理解构造并且使应用程序适于在其各自设施内的实施。在至少一个实施例中，可以选择用于图像重建的应用程序以包括在部署管线4710中，但是由成像设备生成的数据类型可以与在应用程序内使用的数据类型不同。在至少一个实施例中，可以在部署管线4710内使用DICOM适配器4702B(和/或DICOM读取器)或另一数据类型(例如，RIS、CIS、REST兼容、RPC、原始等)适配器或读取器，以将数据转换到部署系统4606内的应用程序可使用的形式。在至少一个实施例中，可以累积和预处理对DICOM、RIS、CIS、REST兼容、RPC、原始和/或其他数据类型库的访问，包括解码、提取和/或执行任何卷积、颜色校正、清晰度、伽玛和/或其他对数据的增强。在至少一个实施例中，DICOM、RIS、CIS、符合REST的RPC、和/或原始数据可以是无序的，并且可以执行预通过以组织或对所收集的数据进行分类。在至少一个实施例中，因为各种应用程序可以共享共同的图像操作，所以在一些实施例中，可以使用数据增强库(例如，作为服务4620之一)来加速这些操作。在至少一个实施例中，为了避免依赖于CPU处理的常规处理方法的瓶颈，并行计算平台4730可以用于这些处理任务的GPU加速。

[0415] 在至少一个实施例中，图像重建应用程序可以包括包括使用机器学习模型的处理任务。在至少一个实施例中，用户可能希望使用他们自己的机器学习模型，或者从模型注册表4624中选择机器学习模型。在至少一个实施例中，用户可以实现他们自己的机器学习模型或选择机器学习模型用于包含在执行处理任务的应用程序中。在至少一个实施例中，应用程序可以是可选的和可定制的，并且通过定义应用程序的构造，针对特定用户的应用程序的部署和实现被呈现为更加无缝的用户体验。在至少一个实施例中，通过利用系统4700的其他特征，例如服务4620和硬件4622，部署管线4710可以对用户更加友好、提供更容易的集成、并产生更准确、有效和及时的结果。

[0416] 在至少一个实施例中，部署系统4606可以包括用户界面4714(例如，图形用户界面、网络界面等)，该用户界面可以被用于选择要包括在部署管线4710中的应用程序，在设置和/或部署期间，布置应用程序，修改或改变应用程序或参数或其构造，在设置和/或部署期间使用部署管线4710并与之交互，和/或以其他方式与部署系统4606交互。对于训练系统4604(未示出)，用户界面4714(或不同的用户界面)可用于选择在部署系统4606中使用的模型，在训练系统4604中选择用于训练或重新训练的模型，和/或用于以其他方式与训练系统4604交互。

[0417] 在至少一个实施例中，除了应用程序编排系统4728之外，还可以使用管线管理器4712来管理部署管线4710的应用程序或容器与服务4620和/或硬件4622之间的交互。在至少一个实施例中，管线管理器4712可以被配置为促进从应用程序到应用程序，从应用程序到服务4620和/或从应用程序或服务到硬件4622的交互。在至少一个实施例中，尽管被示为包括在软件4618中，这并不意味着限制，并且在一些示例中(例如，如图48所示)，管线管理器4712可以包括在服务4620中。在至少一个实施例中，应用程序编排系统4728(例如，Kubernetes、DOCKER等)可以被包括可能包括容器编排系统，该系统可以将应用程序分组为多个容器，作为逻辑单元进行协调、管理、扩展和部署。在至少一个实施例中，通过将来自部署管线4710的应用程序(例如，重建应用程序、分段应用程序等)与各个容器相关联，每个应用程序可以在自包含的环境中(例如，在内核级别下)执行以提高速度和效率。

[0418] 在至少一个实施例中,每个应用程序和/或容器(或其映像)可以单独开发、修改和部署(例如,第一用户或开发人员可以开发、修改和部署第一应用程序和第二用户或开发人员可以开发、修改和部署与第一用户或开发人员分开的第二应用程序),这可以使专注于并关注单个应用程序和/或容器的任务,而不会受到阻碍另一个应用程序或容器的任务。在至少一个实施例中,管线管理器4712和应用编排系统4728可以辅助不同容器或应用程序之间的通信和协作。在至少一个实施例中,只要每个容器或应用程序的预期输入和/或输出是为系统(例如,基于应用程序或容器的构造)所已知的,应用程序编排系统4728和/或管线管理器4712可以促进每个应用程序或容器之中和之间的通信以及资源共享。在至少一个实施例中,由于部署管线4710中的一个或多个应用程序或容器可以共享相同的服务和资源,因此应用程序编排系统4728可以协调、负载平衡并确定各种应用程序之间或容器之间的服务或资源的共享。在至少一个实施例中,调度器可以用于跟踪应用程序或容器的资源需求,这些资源的当前使用或计划使用以及资源可用性。因此,在至少一个实施例中,调度器可以鉴于系统的需求和可用性而将资源分配给不同的应用程序,并且在应用程序之中以及应用程序之间分配资源。在一些示例中,调度器(和/或应用编排系统4728的其他组件)可以基于施加在系统上的约束(例如,用户约束),例如服务质量(QoS)、用于数据输出的需求的紧迫性(例如,确定是执行实时处理还是延迟处理)等,来确定资源可用性和分配。

[0419] 在至少一个实施例中,由部署系统4606中的应用程序或容器利用并由其共享的服务4620可以包括计算服务4716、AI服务4718、可视化服务4720和/或其他服务类型。在至少一个实施例中,应用程序可以调用(例如,执行)服务4620中的一个或多个以执行针对应用程序的处理操作。在至少一个实施例中,应用程序可以利用计算服务4716来执行超级计算或其他高性能计算(HPC)任务。在至少一个实施例中,可以利用计算服务4716来执行并行处理(例如,使用并行计算平台4730),以通过一个或多个应用程序和/或单个应用程序的一个或多个任务来基本同时地处理数据。在至少一个实施例中,并行计算平台4730(例如,NVIDIA的CUDA)可以实现在GPU(GPGPU)(例如,GPU 4722)上的通用计算。在至少一个实施例中,并行计算平台4730的软件层可以提供对GPU的虚拟指令集和并行计算元件的访问,以执行计算内核。在至少一个实施例中,并行计算平台4730可以包括存储器,并且在一些实施例中,可以在多个容器之间和之中,和/或在单个容器内的不同处理任务之间和之中共享存储器。在至少一个实施例中,可以为多个容器和/或容器内的多个过程生成进程间通信(IPC)调用,以使用来自并行计算平台4730的共享存储器段的相同数据(例如,其中一个应用程序或多个应用程序的多个不同阶段正在处理相同的信息)。在至少一个实施例中,不是复制数据并将数据移动到存储器中的不同位置(例如,读/写操作),而是可以将存储器相同位置中的相同数据用于任何数量的处理任务(例如,在同一时间,在不同时间等)。在至少一个实施例中,由于数据被用作处理的结果来生成新数据,因此可以在各种应用程序之间存储和共享数据的新位置的信息。在至少一个实施例中,数据的位置以及更新或修改的数据的位置可以是容器中如何理解有效载荷的定义的一部分。

[0420] 在至少一个实施例中,可以利用AI服务4718来执行推理服务,以执行与应用程序相关联的机器学习模型(例如,被赋予执行应用程序的一个或多个处理任务的任务)。在至少一个实施例中,AI服务4718可以利用AI系统4724来执行机器学习模型(例如,诸如CNN

的神经网络) 以用于分割、重构、对象检测、特征检测、分类和/或其他推理任务。在至少一个实施例中,部署管线4710的应用程序可以使用来自训练系统4604 的输出模型4616中的一个或更多个和/或其他应用程序的模型来对成像数据(例如,DICOM数据、RIS数据、CIS数据、REST兼容的数据、RPC 数据,原始数据等)进行推理。在至少一个实施例中,使用应用编排系统 4728(例如,调度器)进行推理的两个或更多个示例可以是可用的。在至少一个实施例中,第一类别可以包括可以实现更高服务水平协议的高优先级/低等待时间路径,例如用于在紧急情况下对紧急请求执行推理,或者在 诊断期间提供给放射线医生。在至少一个实施例中,第二类别可以包括标准优先级路径,该标准优先级路径可以用于可能不紧急的请求或者可以在稍后的时间执行分析的请求。在至少一个实施例中,应用协调系统4728可以基于用于AI服务4718的不同推理任务的优先级路径来分配资源(例如,服务4620和/或硬件4622)。

[0421] 在至少一个实施例中,共享存储可以被安装到系统4700内的AI服务4718。在至少一个实施例中,共享存储可以用作高速缓存(或其他存储 设备类型),并且可以用于处理来自应用程序的推理请求。在至少一个实施例中,当提交推理请求时,部署系统4606的一组API实例可以接收到 请求,并且可以选择一个或更多个实例(例如,用于最佳配合,用于负载平衡等)以处理请求。在至少一个实施例中,为了处理请求,可以将请求 输入数据库中,可以从模型注册表4624中找到机器学习模型(如果尚未在 缓存中),验证步骤可以确保将适当的机器学习模型加载到高速缓存(例如,共享存储)和/或模型的副本可以保存到高速缓存中。在至少一个实施例中,如果应用程序还没有运行或者没有足够的应用程序的实例,则可以使用调度器(例如,管线管理器4712的调度器)来启动在请求中引用的应用程序。在至少一个实施例中,如果尚未启动推理服务器以执行模型,则可以启动推理服务器。每个模型可以启动任何数量的推理服务器。在至少一个实施例中,在其中推理服务器被集群化的拉模型中,只要负载平衡是有利的,就可以缓存模型。在至少一个实施例中,推理服务器可以被静态地加载在对应的分布式服务器中。

[0422] 在至少一个实施例中,可以使用在容器中运行的推理服务器来执行推理。在至少一个实施例中,推理服务器的实例可以与模型(以及可选地 模型的多个版本)相关联。在至少一个实施例中,如果当接收到对模型执行推理的请求时推理服务器的实例不存在,则可以加载新实例。在至少一个实施例中,当启动推理服务器时,可以将模型传递到推理服务器,使得 可以使用相同的容器来服务于不同的模型,只要推理服务器作为不同的实例运行即可。

[0423] 在至少一个实施例中,在应用程序执行期间,可以接收针对给定应用程序的推理请求,并且可以加载容器(例如,托管推理服务器的实例)(如果尚未加载),以及启动程序可能会被调用。在至少一个实施例中,容器中的预处理逻辑可以对传入的数据进行加载、解码和/或执行任何附加的预处理(例如,使用一个或多个CPU和/或一个或多个GPU)。在至少一个实施例中,一旦准备好数据进行推理,容器就可以根据需要对数据进行推理。在至少一个实施例中,这可以包括对一个图像(例如,手X射线)的单个推理调用,或者可以要求对数百个图像(例如,胸部CT)进行推理。在至少一个实施例中,应用可以在完成之前总结结果,其可以包括但不限于单个置信度得分、像素水平分割、体素水平分割、生成可视化或生成文本以总结发现。在至少一个实施例中,可以为不同的模型或应用分配不同的优先

级。例如,某些模型可能具有实时(TAT<1min)优先级,而其他模型可能具有较低的优先级(例如,TAT<10min)。在至少一个实施例中,可以从请求机构或实体测量模型执行时间,并且该模型执行时间可以包括伙伴网络传递时间以及在推理服务上的执行。

[0424] 在至少一个实施例中,服务4620与推理应用程序之间的请求传输可以被隐藏在软件开发工具包(SDK)的后面,并且可以通过队列来提供健壮的传输。在至少一个实施例中,将经由用于单个应用程序/租户ID组合的API的请求放置在队列中,并且SDK将从队列中拉出请求并将请求提供给应用程序。在至少一个实施例中,可以在SDK从中拾取队列的环境中提供队列的名称。在至少一个实施例中,通过队列的异步通信可能是有用的,因为它可以允许应用程序的任何实例在可用时拾取工作。可以通过队列将结果传送回去,以确保没有数据丢失。在至少一个实施例中,队列还可以提供对工作进行分段的能力,因为最高优先级的工作可以进入与应用程序的大多数实例连接的队列,而最低优先级的工作可以进入与单个实例连接的按接收到的顺序处理任务的队列。在至少一个实施例中,应用可以在云4726中生成的GPU加速的实例上运行,并且推理服务可以在GPU上执行推理。

[0425] 在至少一个实施例中,可视化服务4720可被利用来生成可视化以查看应用程序和/或部署管线4710的输出。在至少一个实施例中,可视化服务4720可利用GPU 4722来生成可视化。在至少一个实施例中,可视化服务4720可以实现诸如射线追踪之类的渲染效果,以生成更高质量的可视化。在至少一个实施例中,可视化可以包括但不限于2D图像渲染、3D体积渲染、3D体积重建、2D断层摄影切片、虚拟现实显示、增强现实显示等。

[0426] 在至少一个实施例中,可以使用虚拟化环境,生成虚拟交互式显示器或环境(例如虚拟环境),以供系统用户(例如医生、护士、放射科医生等)进行交互。在至少一个实施例中,可视化服务4720可以包括内部可视化器、电影和/或其他渲染或图像处理能力或功能(例如,射线追踪、光栅化、内部光学器件等)。

[0427] 在至少一个实施例中,硬件4622可以包括GPU 4722、AI系统4724、云4726和/或用于执行训练系统4604和/或部署系统4606的任何其他硬件。在至少一个实施例中,GPU 4722(例如NVIDIA的TESLA和/或QUADRO GPU)可以包括任意数量的GPU,这些GPU可用于执行计算服务4716、AI服务4718、可视化服务4720、其他服务和/或任何功能或特性的处理任务软件4618。例如,关于AI服务4718,GPU 4722可用于对成像数据(或机器学习模型使用的其他数据类型)执行预处理,对机器学习模型的输出进行后处理和/或执行推理(例如执行机器学习模型)。在至少一个实施例中,云4726、AI系统4724和/或系统4700的其他组件可以使用GPU4722。在至少一个实施例中,云4726可以包括用于深度学习任务的GPU优化平台。在至少一个实施例中,AI系统4724可以使用GPU,并且可以使用一个或多个AI系统4724来执行云4726(或至少一部分负责深度学习或推理的任务)。同样,尽管硬件4622被示出为离散的组件,这并不旨在限制本发明,并且硬件4622的任何组件可以与硬件4622的任何其他组件组合或由其利用。

[0428] 在至少一个实施例中,AI系统4724可以包括被配置用于推理、深度学习、机器学习和/或其他人工智能任务的专用计算系统(例如,超级计算机或HPC)。在至少一个实施例中,除了CPU、RAM、存储器和/或其他组件、功能部件或功能之外,AI系统4724(例如,NVIDIA的DGX)可以包括可以使用多个GPU 4722执行的GPU优化的软件(例如,软件堆栈)。在至少一个实施例中,可以在云4726中(例如,在数据中心中)实现一个或多个AI系统4724,以

执行系统4700的一些或全部基于AI的处理任务。

[0429] 在至少一个实施例中,云4726可以包括GPU加速的基础设施(例如,NVIDIA的NGC),其可以提供用于执行系统4700的处理任务的GPU优化的平台。在至少一个实施例中,云4726可以包括AI系统4724,用于执行系统4700的一个或多个基于AI的任务(例如,作为硬件抽象和缩放平台)。在至少一个实施例中,云4726可以与利用多个GPU的应用编排系统4728集成,以实现应用和服务4620之间和之中的无缝缩放和负载平衡。在至少一个实施例中,如本文所述,云4726可以承担执行系统4700的至少一些服务4620的任务,系统4700包括计算服务4716、AI服务4718和/或可视化服务4720。在至少一个实施例中,云4726可以执行小批量和大批量推理(例如,执行NVIDIA的TENSOR RT),提供加速的并行计算API和平台4730(例如,NVIDIA的CUDA),执行应用编排系统4728(例如,KUBERNETES),提供图形渲染API和平台(例如,用于光线追踪、2D图形、3D图形和/或其他渲染技术以产生更高质量的电影效果),和/或可以为系统4700提供其他功能。

[0430] 在至少一个实施例中,为了保护患者机密性(例如,在备用设备使用患者数据或记录的地方),云4726可以包括注册表-诸如深度学习容器注册表。在至少一个实施例中,注册表可以存储用于对可以对患者数据执行预处理、后处理或其他处理任务的应用的容器进行实例化。在至少一个实施例中,云4726可以接收包括患者数据以及容器中的传感器数据的数据,仅对那些容器中的传感器数据执行所请求的处理,然后将结果输出和/或可视化转发给适当的参与者和/或设备(例如用于可视化或诊断的本地部署医疗设备),而无需提取、存储或访问患者数据。在至少一个实施例中,根据HIPAA和/或其他数据法规来保留患者数据的机密性。

[0431] 图48包括根据至少一个实施例的用于处理成像数据的部署管线4710A的示例说明。在至少一个实施例中,系统4700,特别是部署系统4606,可以用于定制、更新和/或集成部署管线4710A到一个或多个产品环境中。在至少一个实施例中,图48的部署管线4710A可以包括部署管线4710A的非限制性示例,其可以由设施(例如,医院、诊所、实验室、研究环境等)处的特定用户(或用户团队)定制定义。在至少一个实施例中,为了定义用于CT扫描仪4802的部署管线4710A,用户可以例如从容器注册表中选择一个或多个应用程序,这些应用程序执行相对于由CT扫描仪4802生成的成像数据的特定功能或任务。在至少一个实施例中,应用程序可以作为可以利用系统4700的服务4620和/或硬件4622的容器的容器应用到部署管线4710A。此外,部署管线4710A可以包括附加处理任务或应用程序,这些任务或应用程序可以被实现以准备供应用程序使用的数据(例如DICOM适配器4702B和DICOM阅读器4806可在部署管线4710A中使用,以准备供CT重建4808、器官分割4810等使用的数据)。在至少一个实施例中,可以定制或选择部署管线4710A以用于一致的部署,一次使用或另一频率或间隔。在至少一个实施例中,用户可能希望在特定间隔内具有针对几个对象的CT重建4808和器官分割4810,并且因此可以在该时间段内部署管线4710A。在至少一个实施例中,用户可以针对来自系统4700的每个请求选择用户想要针对该请求对该数据执行处理的应用程序。在至少一个实施例中,可以以任何间隔调整部署管线4710A,并且由于系统4700内的容器结构的适应性和可伸缩性,这可以是无缝的过程。

[0432] 在至少一个实施例中,图48的部署管线4710A可以包括CT扫描仪4802,其生成患者或受试者的成像数据。在至少一个实施例中,来自CT扫描仪4802的成像数据可以存储在

与容纳CT扫描仪4802的设施相关联的PACS服务器4804上。在至少一个实施例中,PACS服务器4804可以包括软件和/或可以直接与设施处的成像模态对接的硬件组件(例如CT扫描仪4802)。在至少一个实施例中,DICOM适配器4702B可以使能使用DICOM协议发送和接收DICOM对象。在至少一个实施例中,DICOM适配器4702B可以帮助准备或配置来自PACS服务器4804的DICOM数据,以供部署管线4710A使用。在至少一个实施例中,一旦通过DICOM适配器4702B处理了DICOM数据,管线管理器4712就可以将数据路由到部署管线4710A。在至少一个实施例中,DICOM读取器4806可以从DICOM数据(例如,如在可视化4816A中所示的原始正弦图数据)提取图像文件 和任何相关联的元数据。在至少一个实施例中,被提取的工作文件可以被存储在高速缓存中,以被部署管线4710A中的其他应用程序更快地处理。在至少一个实施例中,一旦DICOM读取器4806完成了数据的提取和/或存储,就可以将完成信号传送到管线管理器4712。在至少一个实施例中,管线管理器4712然后可以发起或调用一个或多个其他部署管线4710A 中的应用程序或容器。

[0433] 在至少一个实施例中,一旦数据(例如,原始正弦图数据)可用于CT重建4808应用的处理,就可以执行CT重建4808应用和/或容器。在至少一个实施例中,CT重建4808可以从高速缓存中读取原始正弦图数据,从原始正弦图数据中重建图像文件(例如,如可视化4816B所示),并且将得到的图像文件存储在高速缓存中。在至少一个实施例中,在重建完成时,可以向管线管理器4712发信号通知重建任务已完成。在至少一个实施例中,一旦重建完成,并且重建的图像文件可以被存储在高速缓存器(或其他存储设备)中,则器官分割4810应用程序和/或容器可以由管线管理器4712触发。器官分割4810应用程序和/或容器可以从高速缓存中读取图像文件,将图像文件规范化或转换为适合推理的格式(例如,将图像文件转换为机器学习模型的输入分辨率),然后运行对标准化的图像的推理。在至少一个实施例中,为了对归一化的图像进行推理,器官分割4810应用程序和/或容器可以依赖服务4620,并且管线管理器4712和/或应用编排系统4728可以通过器官分割4810应用程序和/或容器来促进服务4620的使用。例如,器官分割4810应用程序和/或容器可以利用AI服务4718来对归一化图像执行推理,并且AI服务4718可以利用硬件4622(例如,AI系统4724)来执行AI服务4718。在至少一个实施例中,推理的结果是可以存储在高速缓存(或其他存储设备)中的掩码文件(例如,如可视化4816C所示)。

[0434] 在至少一个实施例中,一旦处理DICOM数据和/或从DICOM数据中提取的数据的应用程序已经完成处理,就可以为管线管理器4712生成信号。在至少一个实施例中,管线管理器4712然后可以执行DICOM写入器4812从高速缓存(或其他存储设备)读取结果,将结果打包成DICOM格式(例如,作为DICOM输出4814),以供生成请求的设施处的用户使用。在至少一个实施例中,然后可以将DICOM输出4814发送到DICOM适配器4702B,以准备DICOM输出4814以存储在PACS服务器4804上(例如,以供设施处的DICOM查看器查看)。在至少一个实施例中,响应于对重建和分段的请求,可视化4816B和4816C可以被生成并且对于用户可用于诊断、研究和/或用于其他目的。

[0435] 尽管图示为在部署管线4710A中的连续应用,但是在至少一个实施例中,可以并行地处理CT重建4808和器官分割4810应用程序。在至少一个实施例中,在应用程序彼此不依赖并且数据可用于每个应用程序的情况下(例如,在DICOM阅读器4806提取数据之后),可以在相同的时间,基本上在相同的时间或者有一些重叠的情况下执行应用程序。在至少

一个实施例中,在两个或更多个应用需要类似服务4620的情况下,系统4700 的调度器可以用于负载平衡并且在各种应用程序之间以及在各个应用程序 之间分配计算或处理资源。在至少一个实施例中,在一些实施例中,并行 计算平台4730可以用于对应用程序执行并行处理,以减少部署管线4710A 的运行时间以提供实时结果。

[0436] 在至少一个实施例中,并参考图49A-图49B所示,部署系统4606 可以被实现为一种或更多种虚拟仪器,以利用成像设备(例如,CT扫描仪, X射线机、MRI机器等)、测序设备、基因组设备和/或其他设备类型执行 不同的功能,例如图像处理、分割、增强、AI、可视化和推理。在至少一个实施例中,系统4700可以允许创建和提供虚拟仪器,该虚拟仪器可以包括软件定义的部署管线4710,该软件定义的部署管线4710可以接收由一个或更多个设备生成的原始/未处理的输入数据并输出已处理的/重建的数据。在至少一个实施例中,表示虚拟仪器的部署管线4710(例如4710A和 4710B)可以例如通过利用机器学习模型将智能实现到管线中,以向系统 提供容器化的推理支持。在至少一个实施例中,虚拟仪器可以执行任何数 量的容器,每个容器包括应用程序的实例。在至少一个实施例中,例如在 需要实时处理的地方,表示虚拟仪器的部署管线4710可以是静态的(例如, 可以设置容器和/或应用程序),而在其他示例中,用于虚拟仪器的容器和 /或应用可以从应用程序或资源池中(例如,在容器注册表中)选择(例如, 基于每个请求)。

[0437] 在至少一个实施例中,系统4700可以被实例化或执行为在例如在与 放射线机、成像设备和/或设施中的其他设备类型相邻或以其他方式与之通 信的情况下的计算系统中的设施处本地部署的一个或多个虚拟仪器。然而, 在至少一个实施例中,可以在本地数据中心(例如,本地部署数据中心) 中和/或云环境(例如,云4726)中的设备本身的计算系统(例如,与成像 设备集成的计算系统)内实例化或执行本地部署安装。在至少一个实施例中, 在一些示例中,可以由超级计算机或其他HPC系统实例化作为虚拟仪 器操作的部署系统4606。在至少一个实施例中,本地部署安装可以允许用 于实时处理的高带宽用途(例如, 通过更高吞吐量的本地通信接口,例如 以太网上的RF)。在至少一个实施例中,在虚拟仪器支持超声设备或其他 成像模态的情况下,实时或近实时处理可能特别有用,在该超声设备或其 他成像模态中,期望或要求立即可视化以进行准确的诊断和分析。在至少 一个实施例中,当本地需求超过本地部署容量或能力时,云计算架构可能 能够动态爆发到云计算服务提供商或其他计算集群。在至少一个实施例中, 如在此关于训练系统4604所描述的,云架构在被实施时可以被调整用于训 练神经网络或其他机器学习模型。在至少一个实施例中,具有适当的训练 管线,机器学习模型在处理来自其支持的设备的附加数据时,可能会不断 学习和改进。在至少一个实施例中,可以使用附加数据、新数据、现有机 器学习模型和/或新的或更新的机器学习模型来持续改进虚拟仪器。

[0438] 在至少一个实施例中,计算系统可以包括本文描述的硬件4622中的一些或全部, 并且硬件4622可以以包括在设备内的多种方式中的任何一种 来分布,作为耦合到设施中的本地数据中心内和/或云4726中的设备和位 于该设备附近。在至少一个实施例中,因为部署系统4606和关联的应用程 序或容器是在软件中创建的(例如,作为应用程序的分立容器化实例), 虚拟仪器的行为、操作和配置以及虚拟仪器生成的输出可以根据需要进行 修改或自定义,而无需改变或更改虚拟仪器支持的设备的原始输出。

[0439] 图49A包括根据至少一个实施例的支持超声设备的虚拟仪器的示例 数据流程图。

在至少一个实施例中,部署管线4710B可以利用系统4700 的服务4620中的一个或更多个。在至少一个实施例中,部署管线4710B 和服务4620可以在本地或在云4726中利用系统的硬件4622。在一个实施例中,虽然未示出,但是过程4900可以由管线管理器4712、应用编排系统4728和/或并行计算平台4730来促进。

[0440] 在至少一个实施例中,过程4900可以包括从超声设备4902接收成像数据。在至少一个实施例中,可以以DICOM格式(或其他格式,例如 RIS、CIS、REST兼容、RPC、原始、等)存储在PACS服务器上,以及可以被系统4700接收,以供通过部署管线4710选择进行处理或定制为虚拟仪器(例如,虚拟超声)以用于超声设备4902。在至少一个实施例中,可以直接从成像设备(例如,超声设备4902)接收成像数据,并通过虚拟仪器对其进行处理。在至少一个实施例中,通信地耦合在成像设备和虚拟仪器之间的换能器或其他信号转换器可以将由成像设备生成的信号数据转换成可以由虚拟仪器处理的图像数据。在至少一个实施例中,原始数据和/或图像数据可以被应用于DICOM读取器4806,以提取数据以供部署管线4710B的应用程序或容器使用。在至少一个实施例中,DICOM读取器4806 可以利用数据增强库4914(例如,NVIDIA的DALI)作为服务4620(例如,作为计算服务4716之一)来提取、调整大小、重新缩放和/或以其他方式准备数据以供应用程序或容器使用。

[0441] 在至少一个实施例中,一旦准备好数据,就可以执行重建4906应用程序和/或容器以将来自超声设备4902的数据重建为图像文件。在至少一个实施例中,在重建4906之后或与重建4906同时,可以执行检测4908 应用程序和/或容器以用于异常检测、对象检测、特征检测和/或与数据有关的其他检测任务。在至少一个实施例中,可以在检测4908期间使用在重建 4906期间生成的图像文件来识别异常、对象、特征等。在至少一个实施例中,检测4908应用程序可以利用推理引擎4916(例如,作为AI服务4718 之一),以对数据执行推理以生成检测。在至少一个实施例中,检测4908 应用程序可以执行或调用一个或更多个机器学习模型(例如,来自训练系统4604)。

[0442] 在至少一个实施例中,一旦重构4906和/或检测4908完成,从这些 应用程序和/或容器输出的数据可用于生成可视化4910,例如显示在工作站或显示终端上的可视化4912(例如,灰度输出)。在至少一个实施例中,可视化可以允许技术人员或其他用户可视化相对于超声设备4902的部署 管线4710B的结果。在至少一个实施例中,可以通过利用系统4700的渲染组件4918来执行可视化4910(例如,可视化服务4720之一)。在至少一个实施例中,渲染组件4918可以执行2D、OpenGL或光线追踪服务以生成可视化4912。

[0443] 图49B包括根据至少一个实施例的支持CT扫描仪的虚拟仪器的示例数据流程图。在至少一个实施例中,部署管线4710C可利用系统4700的 服务4620中的一个或更多个。在至少一个实施例中,部署管线4710C和 服务4620可在本地或在云4726中利用系统的硬件4622。在至少一个实施例中,尽管未示出,但是过程4920可以由管线管理器4712、应用编排系统4728和/或并行计算平台4730来促进。

[0444] 在至少一个实施例中,过程4920可以包括生成原始数据的CT扫描仪4922,该原始数据可以由DICOM读取器4806接收(例如,在处理等之后直接经由PACS服务器4804接收)。在至少一个实施例中,虚拟CT(由部署管线4710C实例化)可以包括第一实时管线,用于监视患者(例如,患者运动检测AI 4926)和/或用于调整或优化CT扫描仪4922的暴露。(例如,使用曝光控制AI 4924)。在至少一个实施例中,一个或更多个应用程序(例如4924和4926)

可以利用服务4620,例如AI服务4718。在至少一个实施例中,曝光控制AI 4924应用程序(或容器)的输出和/或患者运动检测AI 4926应用程序(或容器)可以用作对CT扫描仪4922和/或技术人员的反馈,以调整曝光(或CT扫描仪4922的其他设置)和/或通知患者减少运动。

[0445] 在至少一个实施例中,部署管线4710C可以包括用于分析由CT扫描仪4922生成的数据的非实时管线。在至少一个实施例中,第二管线可以包括CT重建4808应用程序和/或容器、粗略检测AI 4928应用程序和/或容器、精细检测AI 4932应用程序和/或容器(例如,通过粗略检测AI 4928检测到某些结果的情况)、可视化4930应用程序和/或容器以及DICOM编写器4812(和/或其他数据类型编写器,例如RIS、CIS、REST兼容、RPC、原始等)应用程序和/或容器。在至少一个实施例中,由CT扫描仪4922产生的原始数据可以通过部署管线4710C的管线(被实例化为虚拟CT仪器)以产生结果。在至少一个实施例中,来自DICOM写入器4812的结果可以被发送用于显示和/或可以被存储在PACS服务器4804上以供技术人员、从业者或其他用户以后检索、分析或显示。

[0446] 图50A示出了根据至少一个实施例的用于训练、重新训练或更新机器学习模型的过程5000的数据流程图。在至少一个实施例中,可以使用作为非限制性示例的图47的系统4700来执行过程5000。在至少一个实施例中,过程5000可以利用本文所描述的系统4700的服务4620和/或硬件4622。在至少一个实施例中,由过程5000生成的改进的模型5012可以由部署系统4606针对部署管线4710中的一个或多个容器化应用程序执行。

[0447] 在至少一个实施例中,模型训练4614可以包括使用新的训练数据(例如,新的输入数据,诸如客户数据集5006和/或与输入数据相关联的新的地面实况数据)对初始模型5004(例如,预训练模型)进行重新训练或更新。在至少一个实施例中,为了重新训练或更新初始模型5004,可以重置或删除初始模型5004的输出或损失层,和/或用更新的或新的输出或损失层代替。在至少一个实施例中,初始模型5004可以具有从先前训练中保留的先前精细调整的参数(例如权重和/或偏差),因此训练或重新训练4614可能不需要花费与训练模型从头开始一样长的时间或不需要那么多的处理。在至少一个实施例中,在模型训练4614期间,通过重置或替换初始模型5004的输出或损失层,可以基于与输出的准确性相关联的损失计算或生成关于新的客户数据集5006(例如,图46的图像数据4608)的预测时的损失层来为新数据集更新和重新调整参数。

[0448] 在至少一个实施例中,可以将预训练的模型4706存储在数据存储或注册表中(例如,图46的模型注册表4624)。在至少一个实施例中,预训练模型4706可能已经至少部分地在除设施执行过程5000之外的一个或多个设施处训练。在至少一个实施例中,为了保护患者、受试者、或不同设施的客户,预训练的模型4706可能已经本地部署使用在本地部署生成的客户或患者数据进行了训练。在至少一个实施例中,可以使用云4726和/或其他硬件4622来训练预训练的模型4706,但是机密的、受隐私保护的 患者数据可以不被传送到云4726的任何组件、不由云4726的任何组件使用或云4726的任何组件(其他非本地硬件)不可访问。在至少一个实施例中,在使用来自一个以上设施的患者数据来训练预训练模型4706的情况下,在对来自另一设施的患者或客户数据进行训练之前,可以针对每个设施分别对预训练模型4706进行训练。在至少一个实施例中,例如在客户或患者数据已发布隐私问题(例如,通过放弃、用于实验用途等),或客户或患者数据包含在公共数据集中的情况

下,来自任何数量的设施的客户端或患者数据可以用于在例如数据中心或其他云计算基础设施中的本地部署和/或 备用设备情况下训练预先训练的模型4706。

[0449] 在至少一个实施例中,当选择用于在部署管线4710中使用的应用程序时,用户还可以选择要用于特定应用程序的机器学习模型。在至少一个 实施例中,用户可能不具有要模型以供使用,因此用户可以选择要与应用程序一起使用的预训练模型4706。在至少一个实施例中,对未优化预训练 模型4706进行优化以在用户设施的客户端数据集5006上生成准确结果(例如,基于患者多样性、人口统计学、所使用的医学成像设备的类型等)。

[0450] 在至少一个实施例中,在将预训练的模型4706部署到部署管线4710中以 与一个或更多个应用程序一起使用之前,可以对预训练的模型4706进行更新、重新训练和/或微调以用于相应的设施。

[0451] 在至少一个实施例中,用户可以选择要被更新、重新训练和/或微调 的预训练模型4706,并且预训练模型4706可以被称为用于处理5000中的 训练系统4604的初始模型5004数据。在至少一个实施例中,客户端数据集 5006(例如,成像数据、基因组数据、测序数据或由设施中的设备生成的 其他数据类型)可以用于在初始模型5004上执行模型训练4614(其可以 包括,但不限于,转移学习)以生成细化模型5012。在至少一个实施例中,可以由训练系统4604生成对应于客户端数据集5006的地面实况数据。在至 少一个实施例中,可以至少部分地由临床医生、科学家、医生、从业人员 在设施中生成地面实况数据。(例如,如图46中标记的临床数据4612)。

[0452] 在至少一个实施例中,在一些示例中可以使用AI辅助注释4610来 生成地面实况数据。在至少一个实施例中,AI辅助注释4610(例如,使用 AI辅助注释SDK实现)可以利用机器学习模型(例如,神经网络)来生 成针对客户端数据集的建议或预测的地面实况数据。在至少一个实施例中, 用户5010可以在计算设备5008上的用户界面(图形用户界面(GUI)内 使用注释工具。

[0453] 在至少一个实施例中,用户5010可以经由计算设备5008与GUI交 互以编辑或微调(自动)注释。在至少一个实施例中,多边形编辑特征可 以用于将多边形的顶点移动到更精确或微调的位置。

[0454] 在至少一个实施例中,一旦客户端数据集5006具有相关联的地面实况 数据,则地面实况数据(例如,来自AI辅助注释、手动标记等)可以在模 型训练4614期间用于生成精炼模型5012。在至少一个实施例中,客户端数 据集5006可以被多次应用到初始模型5004,并且地面实况数据可以被用 于更新初始模型5004的参数,直到对于细化模型5012达到可接受的 精度 水平。在至少一个实施例中,一旦生成了细化模型5012,就可以在用于执 行关于医学成像数据的一个或更多个处理任务的设施处的一个或更多个部 署管线4710内部署细化模型5012。

[0455] 在至少一个实施例中,可以将细化模型5012上传到模型注册表4624 中的预训练模型4706,以由另一设施选择。在至少一个实施例中,他的过 程可以在任意数量的设施处完成,使得可以在新数据集上任意次数地进一 步完善细化模型5012,以生成更通用的模型。

[0456] 图50B是根据至少一个实施例的,利用预训练的注释模型来增强注 释工具的客户端-服务器架构5032的示例说明。在至少一个实施例中,可 以基于客户端-服务器架构5032

来实例化AI辅助的注释工具5036。在至少一个实施例中,成像应用程序中的注释工具5036可以帮助放射线医生,例如,识别器官和异常。在至少一个实施例中,成像应用程序可以包括软件工具,作为非限制性示例,该软件工具帮助用户5010识别原始图像5034(例如,在3D MRI或CT扫描中)的特定感兴趣器官上的一些极端点,并接收特定器官的所有2D切片的自动注释结果。在至少一个实施例中,结果可以作为训练数据5038被存储在数据存储器中,并且被用作(例如但不限于)用于训练的地面实况数据。在至少一个实施例中,当计算设备5008发送用于AI辅助注释4610的极限点时,深度学习模型例如可以接收该数据作为输入并返回分割器官或异常的推理结果。在至少一个实施例中,预先实例化的注释工具(例如图50B中的AI辅助注释工具5036B),可以通过对诸如注释辅助服务器5040之类的服务器进行API调用(例如,API调用5044)来增强,例如,注释辅助服务器5040可以包括存储在注释模型注册表中的一组预训练模型5042。在至少一个实施例中,注释模型注册表可以存储预训练的模型5042(例如,机器学习模型,诸如深度学习模型),其被预训练以对特定器官或异常执行AI辅助注释。在至少一个实施例中,可以通过使用训练管线4704来进一步更新这些模型。在至少一个实施例中,随着添加新的标记的临床数据4612,可以随时间改善预安装的注释工具。

[0457] 推理和/或训练逻辑1015用于执行与一个或更多个实施例相关的推理和/或训练操作。本文结合图9和/或图10提供关于推理和/或训练逻辑1015的细节。在至少一个实施例中,推理和/或训练逻辑1015可以用于推理或预测软件程序以使用优化的决策树解决问题。在至少一个实施例中,推理和/或训练逻辑1015可以用于选择用于解决特定操作的内核。

[0458] 其他变型在本公开的精神内。因此,尽管所公开的技术易于进行各种变构和替代构造,但是某些示出的实施例由此在附图中示出并且已经在上面进行了详细描述。然而,应当理解,无意将本发明限制为所公开的特 定形式或形式,相反,其意图是涵盖落入本发明的精神和范围内的所有修改、替代构造和等同形式,如所附权利要求所定义的。

[0459] 在描述所公开的实施例的上下文中(特别是在所附权利要求的上下文中)术语“一”、“一个”和“所述”以及类似指代的使用应被解释为涵盖单数和复数并且不作为术语的定义,除非本文另有说明或与上下文明显矛盾。除非另外指出,否则术语“包括”、“具有”、“包含”和“含有”应解释为开放式术语(即,意思是“包括但不限于”)。术语“连接”(未经修改且指的是物理连接),应理解为完全或部分地包含在,附加到或连接在一起,即使有某物介入。本文中数值范围的引用仅旨在用作一种简写方法,除非本文另有说明,否则分别指代落入该范围内的每个单独值,并且每个单独值都被并入说明书中,就如同在此单独引用一样。术语“集合”(例如,“项目的集合”)或“子集”的使用,除非上下文另有说明或与之矛盾,否则应解释为包含一个或更多个成员的非空集合。此外,除非上下文另有说明或与之矛盾,否则相应集合的术语“子集”不是必需表示相应集合的适当子集,但是该子集和相应集合可以相等。

[0460] 连接的语言,例如“A、B和C中的至少一个”或“A、B和C至少一个”形式的短语,除非另有明确说明或与上下文明显矛盾否则,否则可以作为通常使用与上下文一起理解以呈现项目、条款等,可以是A或B或C,也可以是A和B以及C的集合的任何非空子集。例如,在具有三个成员的集合的示例性示例中,连接短语“A、B和C中的至少一个”和“A、B和C中的

至少一个”是指以下任意集合： $\{A\}$ 、 $\{B\}$ 、 $\{C\}$ 、 $\{A,B\}$ 、 $\{A,C\}$ 、 $\{B,C\}$ 、 $\{A,B,C\}$ 。因此，这种连接语言通常并不旨在暗示某些实施例需要至少一个A、至少一个B和至少一个C，他们每一个用于呈现。另外，除非另有说明或与上下文矛盾，否则术语“多个”表示复数的状态（例如，“多个项目”表示复数个项目）。多个是至少两个项目，但是当明确地或通过上下文指示时可以是更多。此外，除非另有说明或从上下文中另外可知，否则短语“基于”是指“至少部分基于”而不是“仅基于”。

[0461] 可以以任何合适的顺序来执行本文描述的过程的操作，除非本文另有指示或与上下文明显矛盾。在一个实施例中，诸如本文描述的那些过程（或其变型和/或组合）的过程在一个或多个计算机系统中的一个控制下通过硬件或其组合执行，一个或多个计算机系统配置有可执行指令并且被实现为在一个或多个处理器上共同执行的代码（例如，可执行指令、一个或多个计算机程序或一个或多个应用程序）。在一个实施例中，代码以计算机程序的形式存储在计算机可读存储介质上，该计算机程序包括可由一个或多个处理器执行的多个指令的计算机程序。在一个实施例中，计算机可读存储介质是非暂时性计算机可读存储介质，其不包括暂时信号（例如，传播的瞬态电或电磁传输）但包括瞬时信号的收发器内的非暂时性数据存储电路（例如，缓冲器、高速缓存和队列）。在一个实施例中，代码（例如，可执行代码或源代码）被存储在其上存储有可执行指令（或其他存储器以存储可执行指令）的一组一个或多个非暂时性计算机可读存储介质上，该可执行指令在被计算机系统的一个或多个处理器执行时（例如，作为被执行的结果），使计算机系统执行本文所述的操作。在一个实施例中，该组非暂时性计算机可读存储介质包括多个非暂时性计算机可读存储介质，以及多个非暂时性计算机可读存储介质中的一个或多个单独的、非暂时性存储介质缺少全部代码，而多个非暂时性计算机可读存储介质共同存储所有代码。在一个实施例中，可执行指令被执行，使得不同的指令被不同的处理器执行—例如，非暂时性计算机可读存储介质存储指令，并且主CPU执行一些指令，而图形处理器单元执行其他指令。在一个实施例中，计算机系统的不同组件具有独立处理器，以及不同处理器执行指令的不同子集。

[0462] 因此，在一个实施例中，计算机系统配置成实现单独或共同执行本文所述的过程的操作的一个或多个服务，并且这样的计算机系统配置有能够使操作的执行的适用的硬件和/或软件。此外，实现本公开的实施例的计算机系统是单个设备，并且在另一实施例中，是一种分布式计算机系统，其包括以不同方式操作的多个设备，使得该分布式计算机系统执行本文所述的操作，并且使得单个设备不执行所有操作。

[0463] 除非另外要求，本文提供的任何和所有示例或示例性语言（例如，“诸如”）的使用仅旨在更好地阐明本发明的实施例，而不对本发明的范围构成限制。本说明书中的语言不应解释为表示任何未要求保护的要素对于实施本发明是必不可少的。

[0464] 本文引用的所有参考文献，包括出版物、专利申请和专利，均以引用的方式并入本文，其程度如同每个参考文献被单独且具体地指示以引用方式并入本文一样，并在此全文进行阐述。

[0465] 在说明书和权利要求书中，可以使用术语“耦合”和“连接”及其派生词。应当理解，这些术语可能不旨在作为彼此的同义词。相反，在特定示例中，“连接”或“耦合”可用于指示两个或多个元件彼此直接或间接触物理或电接触。“耦合”也可能意味着两个或多个

元素彼此不直接接触，但仍彼此协作或交互。

[0466] 除非另有说明，否则应理解，在整个说明书中，诸如“处理”、“计算处理”、“计算”、“确定”等术语均指计算机或计算系统或类似的电子计算设备的动作和/或过程，这些电子设备将在计算系统的寄存器和/或存储器中表示为物理量（例如电子）的数据（例如电子）操作和/或转换为类似地表示为计算系统的存储器、寄存器或其他此类信息存储器、传输或显示设备中的物理量的其他数据。

[0467] 在类似的方式中，术语“处理器”可以指处理来自寄存器和/或存储器的电子数据并将该电子数据转换成可以存储在寄存器和/或存储器中的其他电子数据的任何设备或设备的一部分。作为非限制性示例，“处理器”可以是中央处理单元（CPU）或图形处理单元（GPU）。“计算平台”可以包括一个或更多个处理器。如本文所使用的，“软件”过程可以包括例如随时间执行工作的软件和/或硬件实体，诸如任务、线程和智能代理。同样，每个过程可以指代多个过程，以依次或并行，连续或间歇地执行指令。术语“系统”和“方法”在本文中可互换使用，以达到一种该系统可以体现一种或更多种方法并且该方法可以被认为是系统的程度。

[0468] 在本文档中，可以参考获得、获取、接收或将模拟或数字数据输入子系统、计算机系统或计算机实现的机器。可以以多种方式来完成获得、获取、接收或输入模拟和数字数据的过程，例如通过接收作为函数调用或对应用程序接口的调用的参数的数据。在一些实施方式中，获得、获取、接收或输入模拟或数字数据的过程可以通过经由串行或并行接口传输数据来完成。在另一个实施方式中，获得、获取、接收或输入模拟或数字数据的过程可以通过经由计算机网络将数据从提供实体转移到获取实体来完成。也可以参考提供、输出、传输、发送或呈现模拟或数字数据。在各种示例中，提供、输出、传输、发送或呈现模拟或数字数据的过程可以通过将数据作为函数调用的输入或输出参数、应用程序编程接口或进程间通信机制的参数进行传输来完成。

[0469] 尽管以上讨论阐述了所描述的技术的示例实施方式，但是其他架构可以用于实现所描述的功能，并且意图在本公开的范围内。此外，尽管出于讨论目的在上面定义了具体的职责分配，但是根据情况，各种功能和职责可能以不同的方式分配和划分。

[0470] 此外，尽管已经以结构特征和/或方法动作专用的语言描述了主题，但应理解，所附权利要求书中定义的主题不必限于所描述的特定特征或动作。而是，将特定特征和动作公开为实现权利要求的示例性形式。

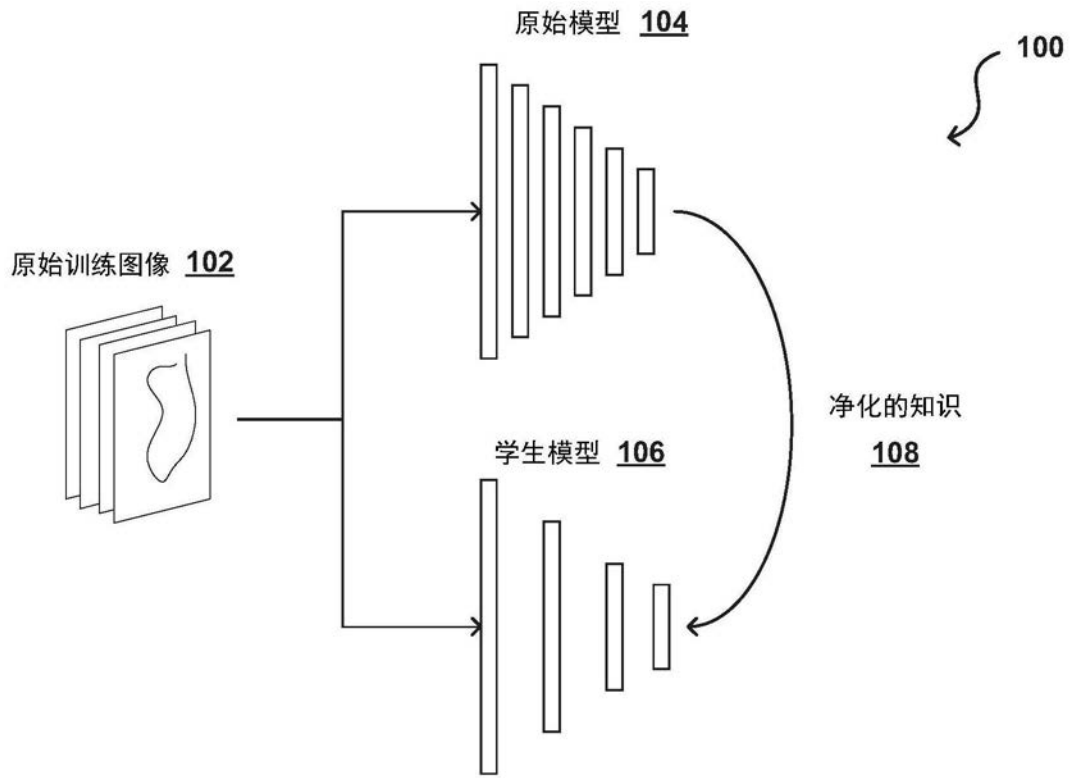


图1A

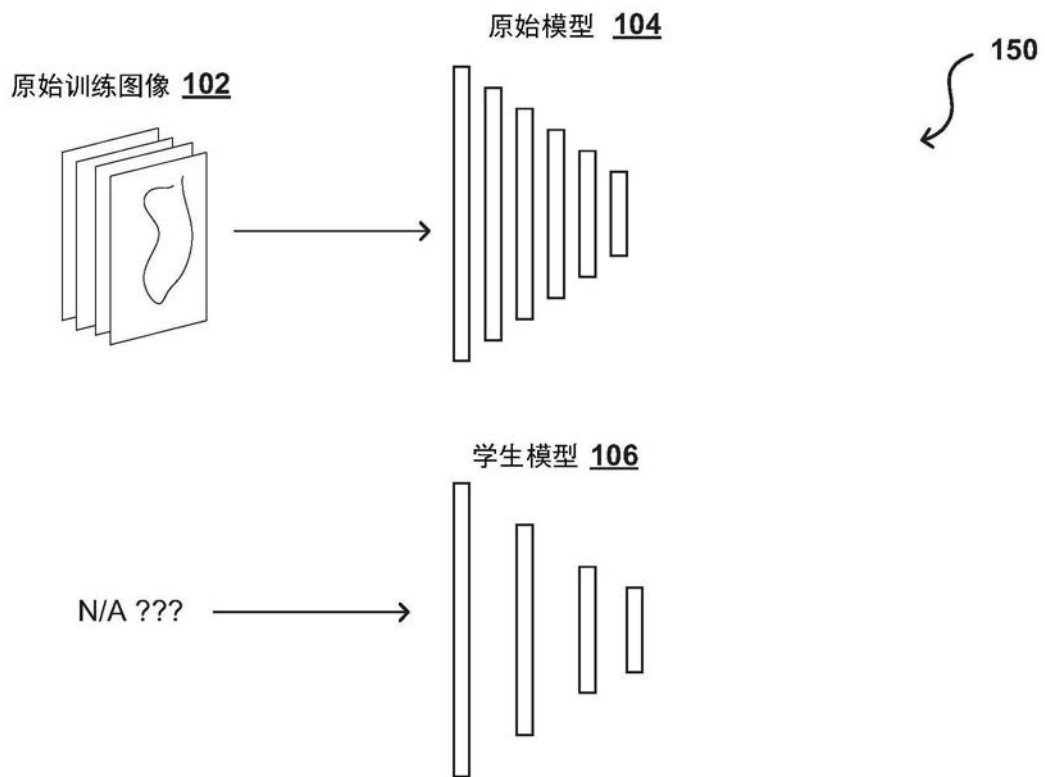


图1B

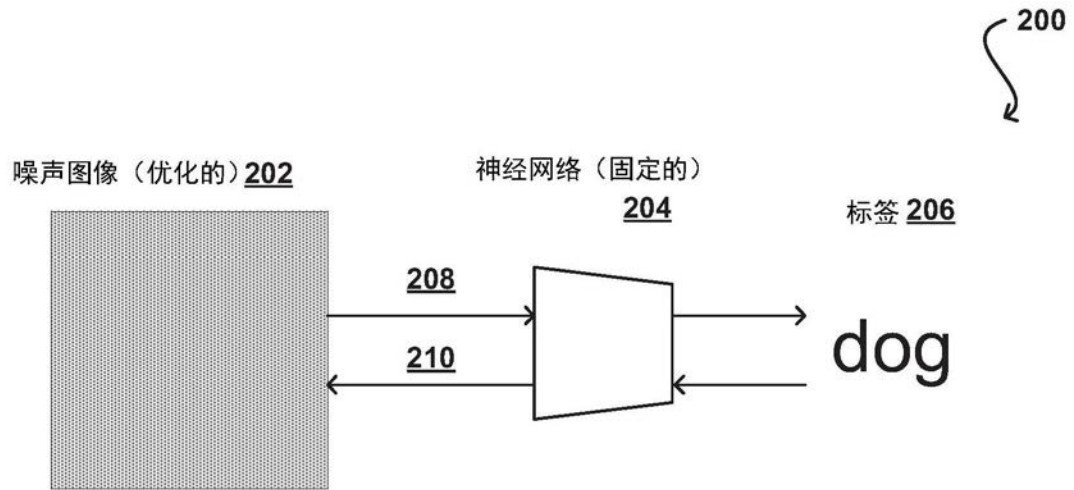


图2A

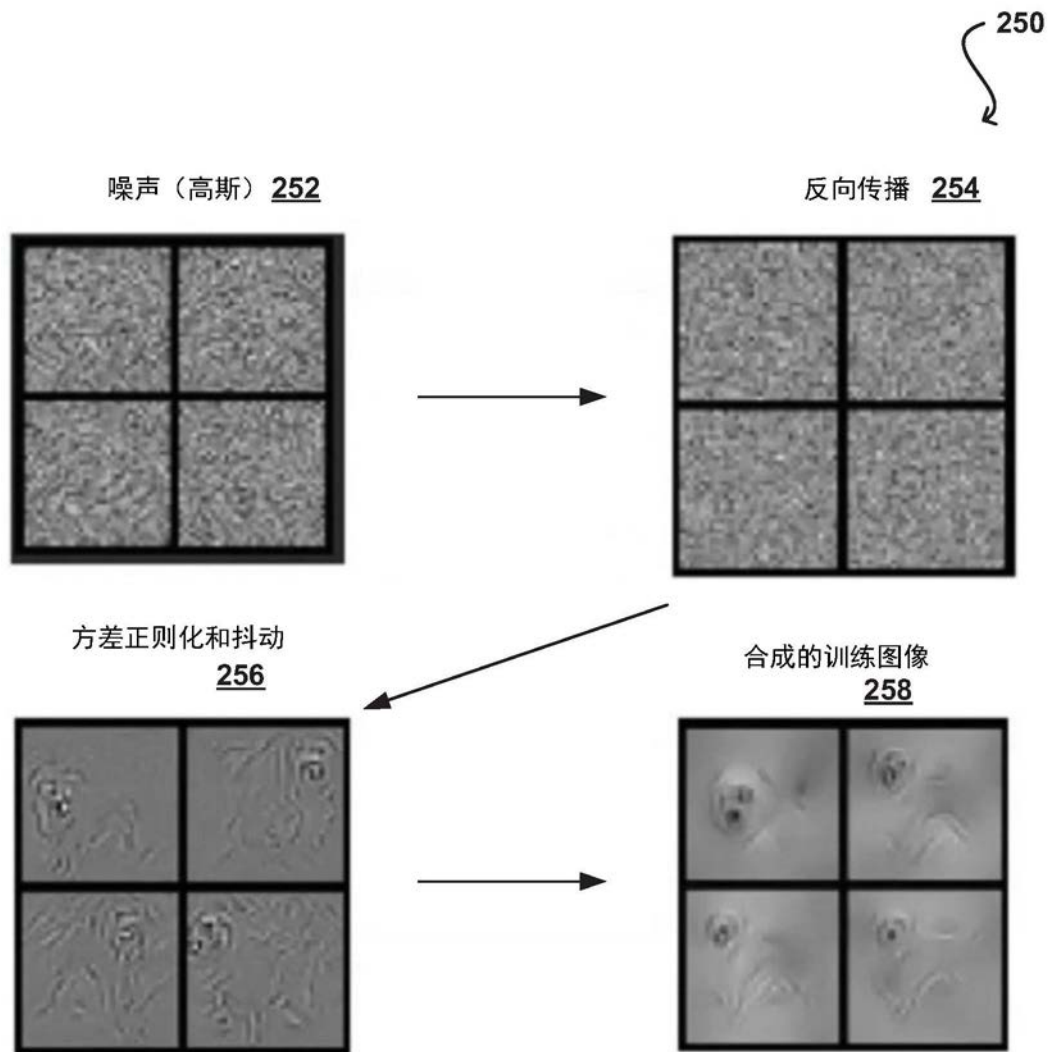


图2B

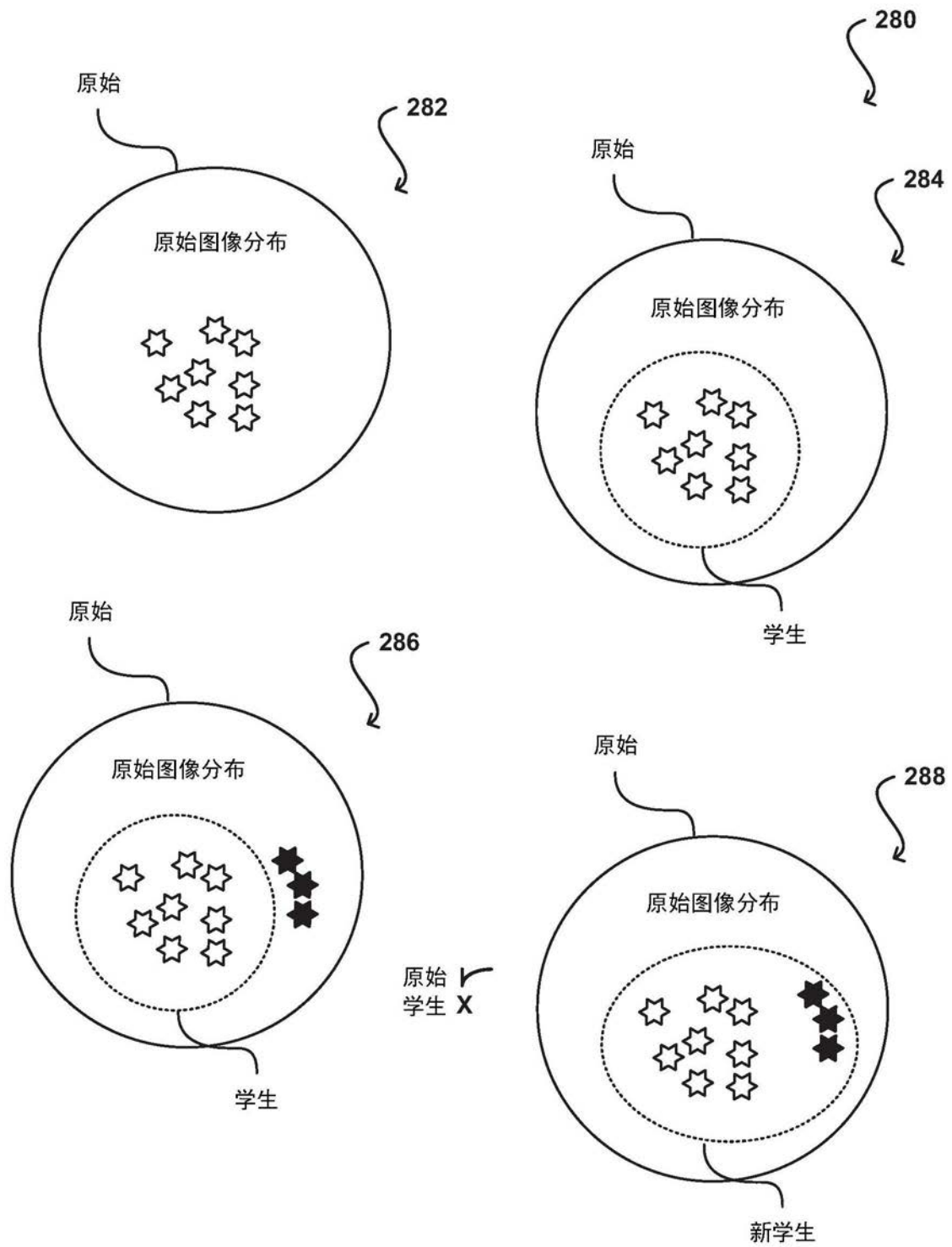


图2C

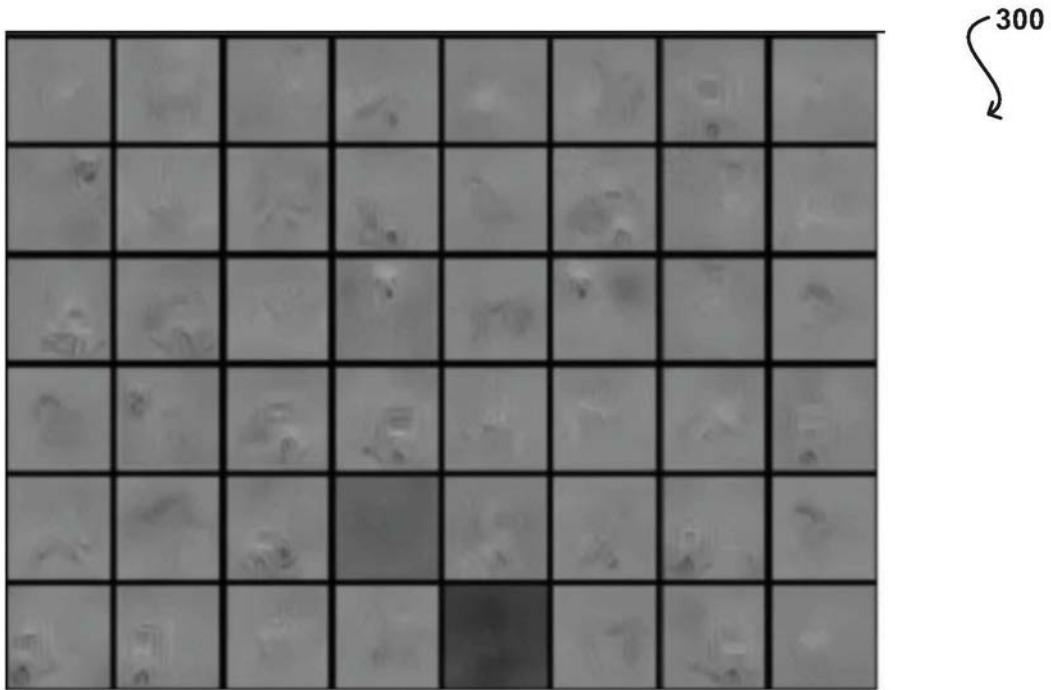


图3A



图3B

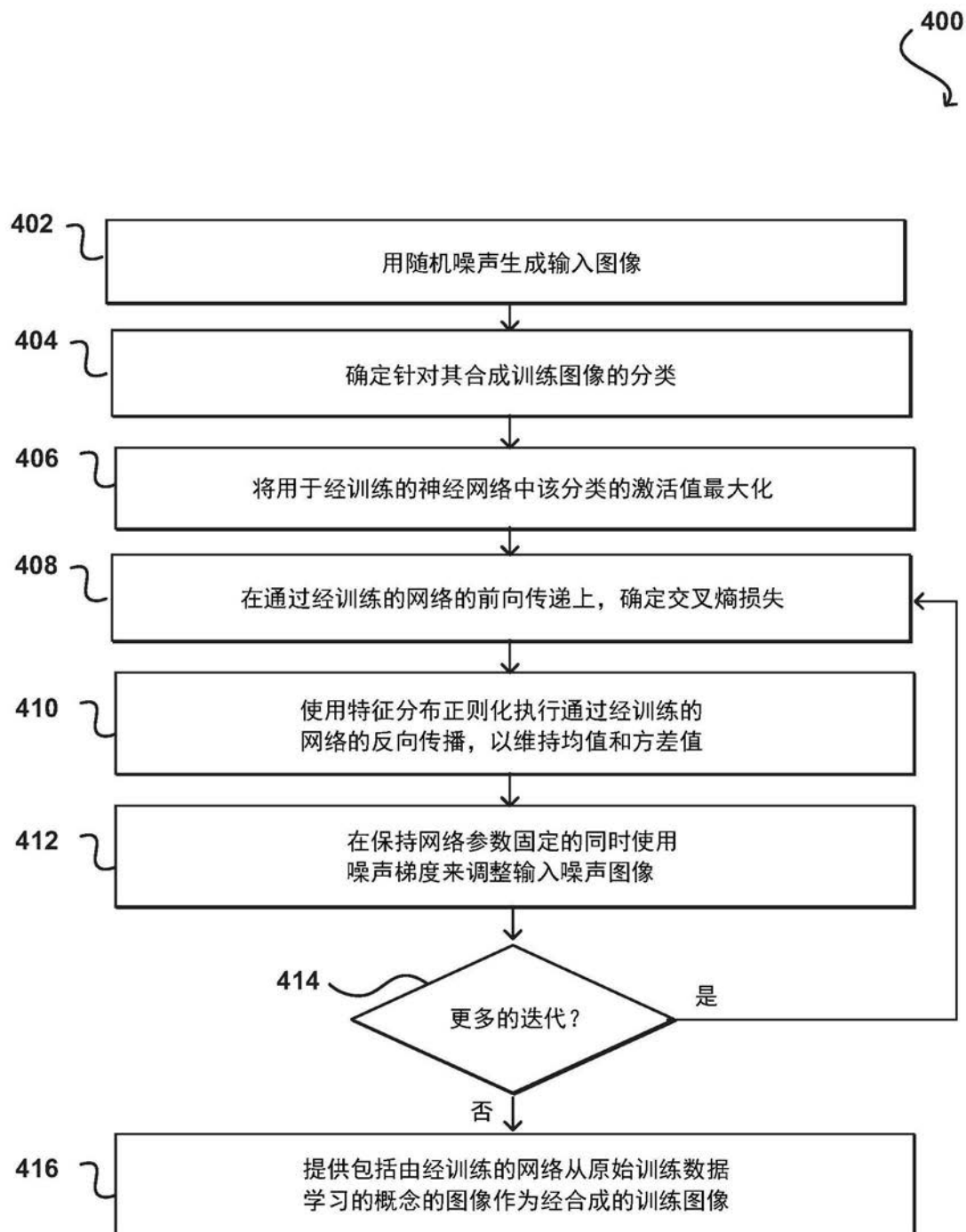


图4

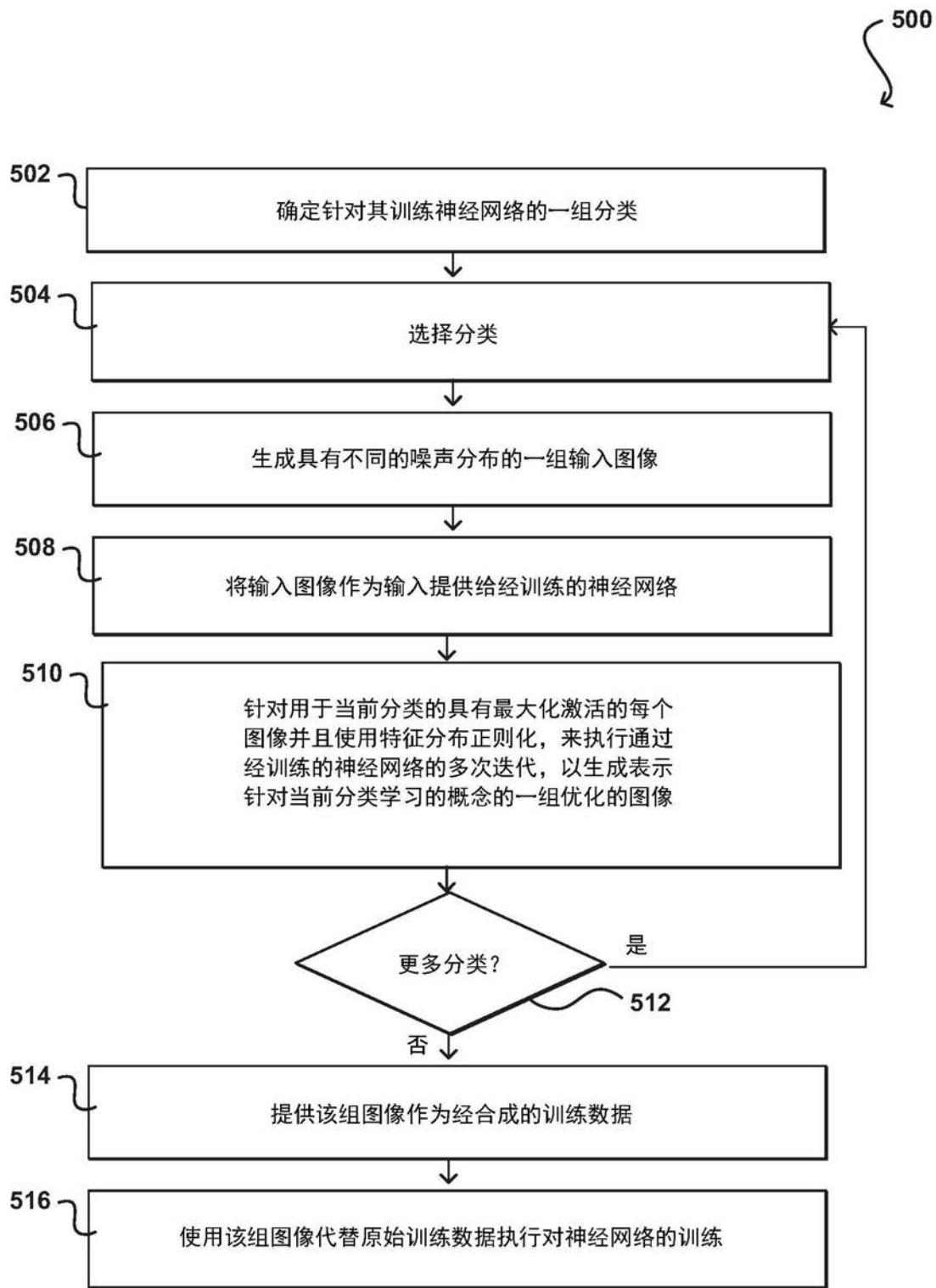


图5

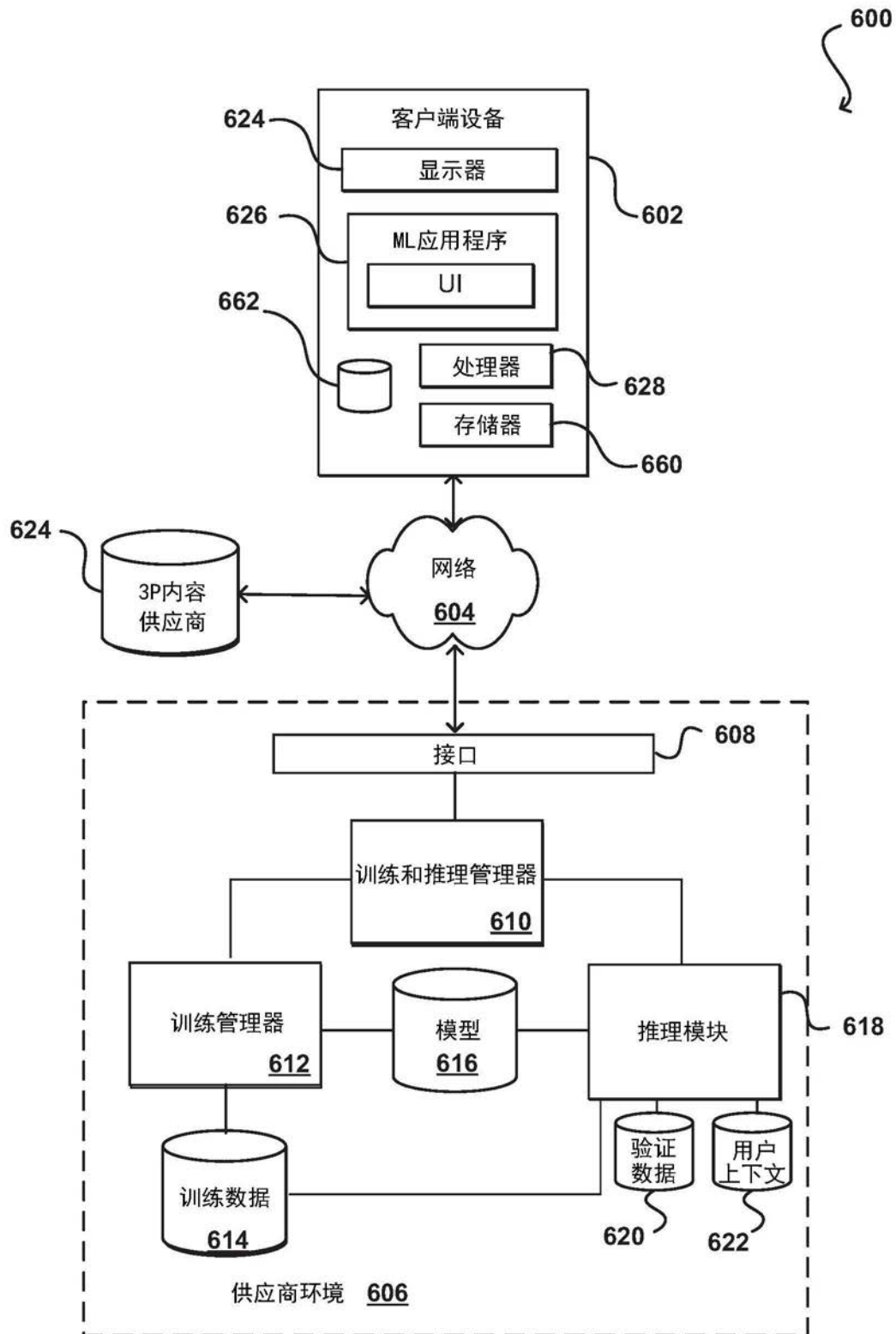


图6

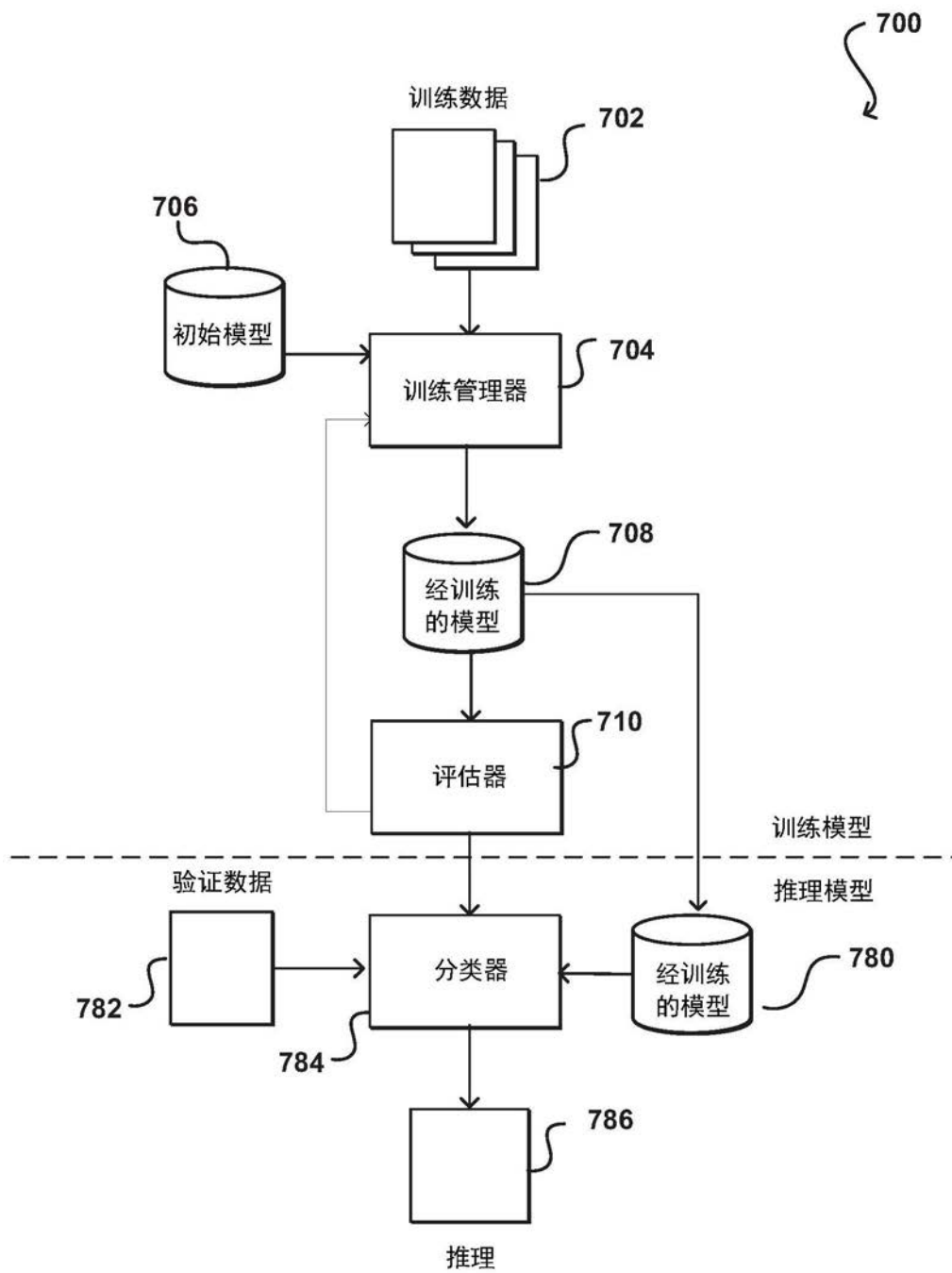


图7

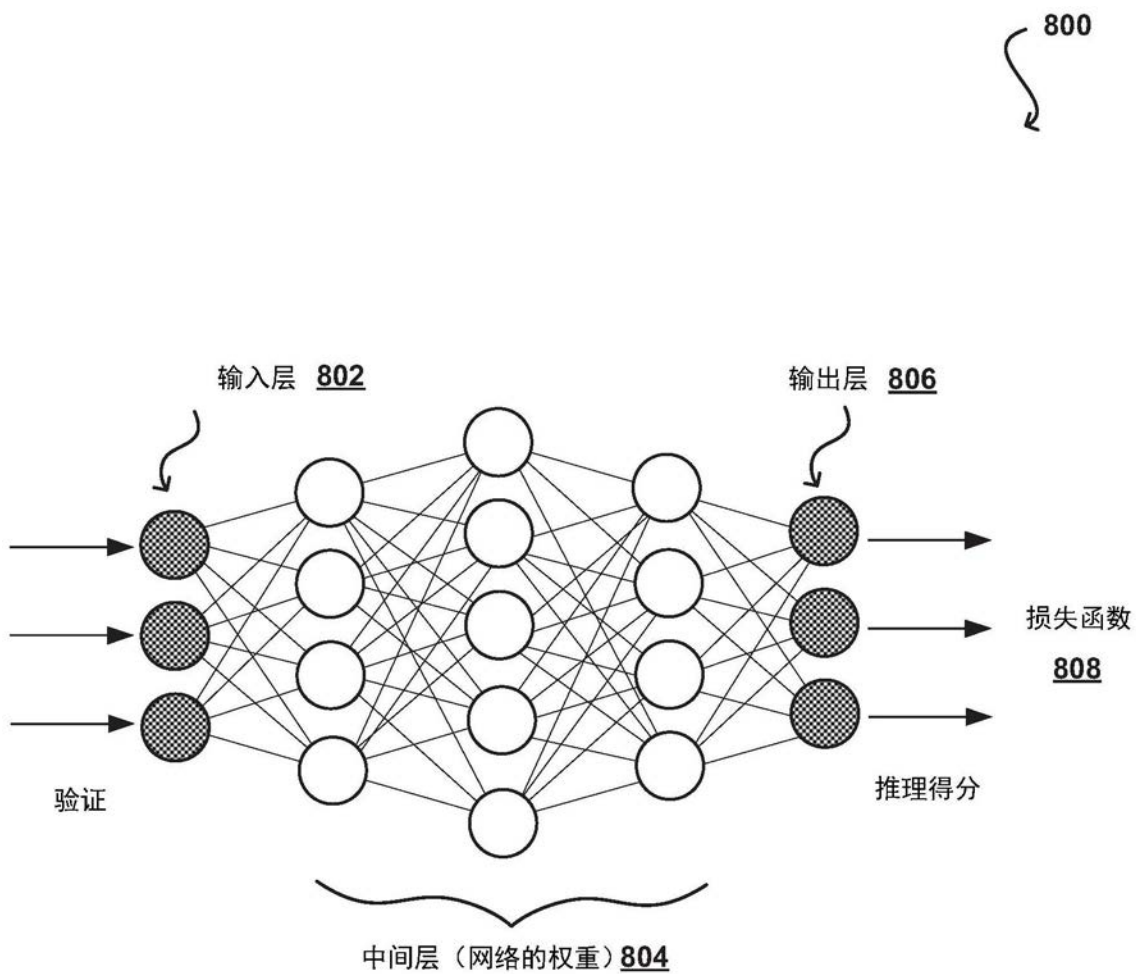


图8

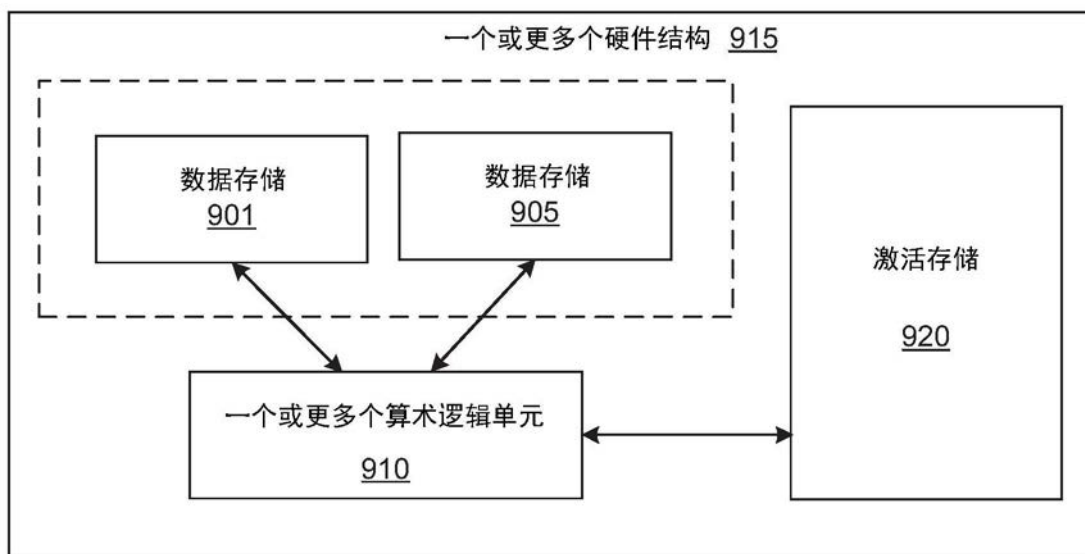


图9

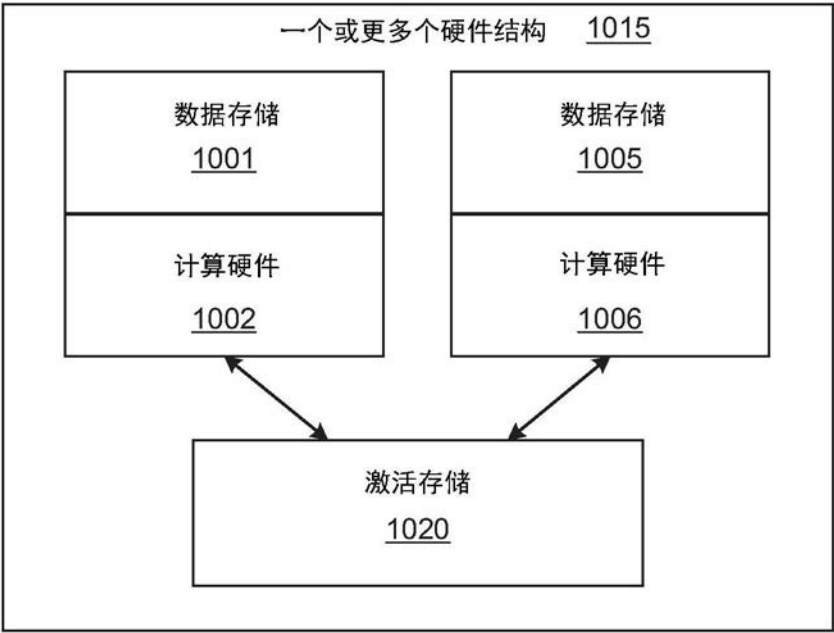


图10

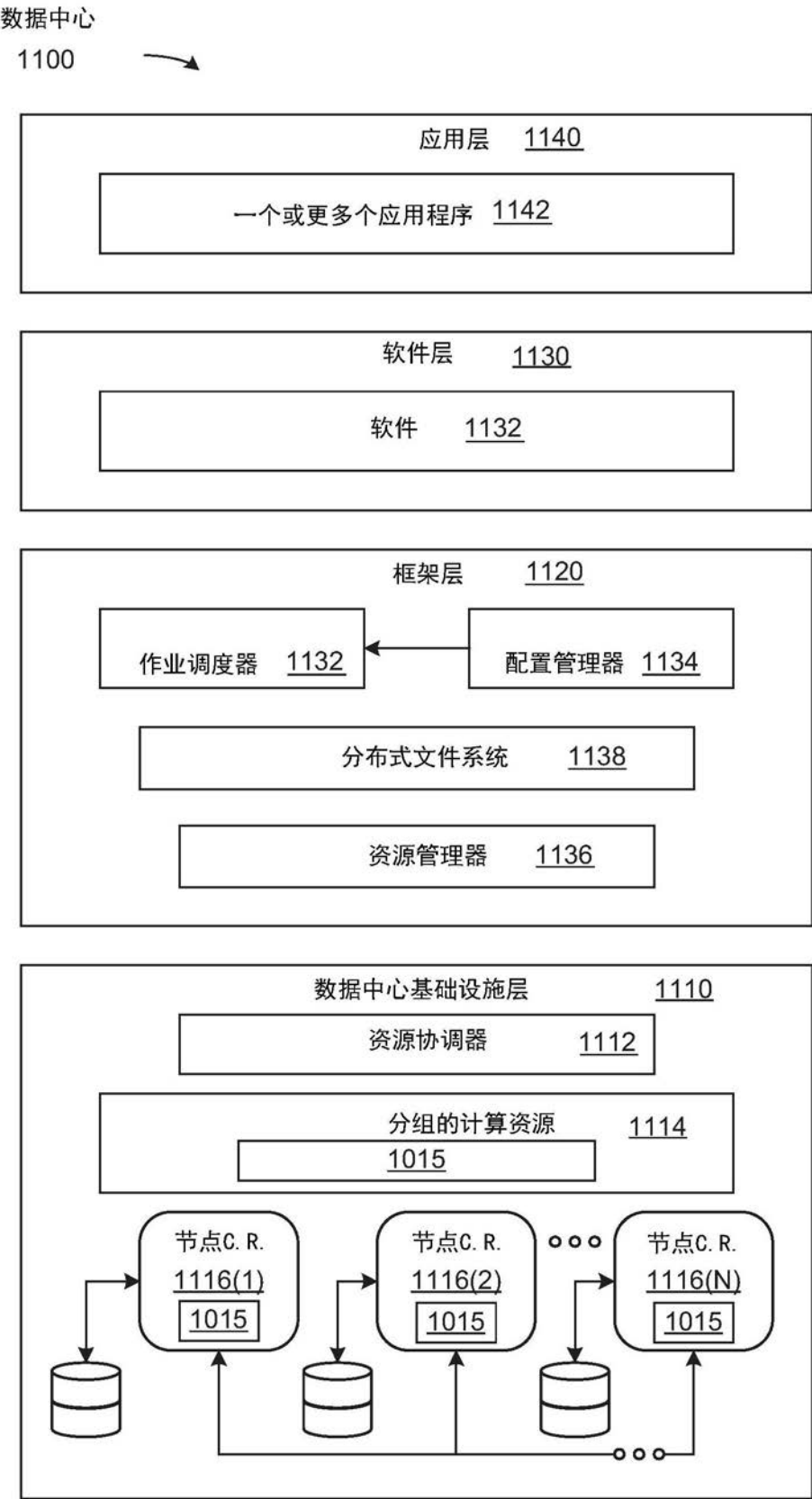


图11

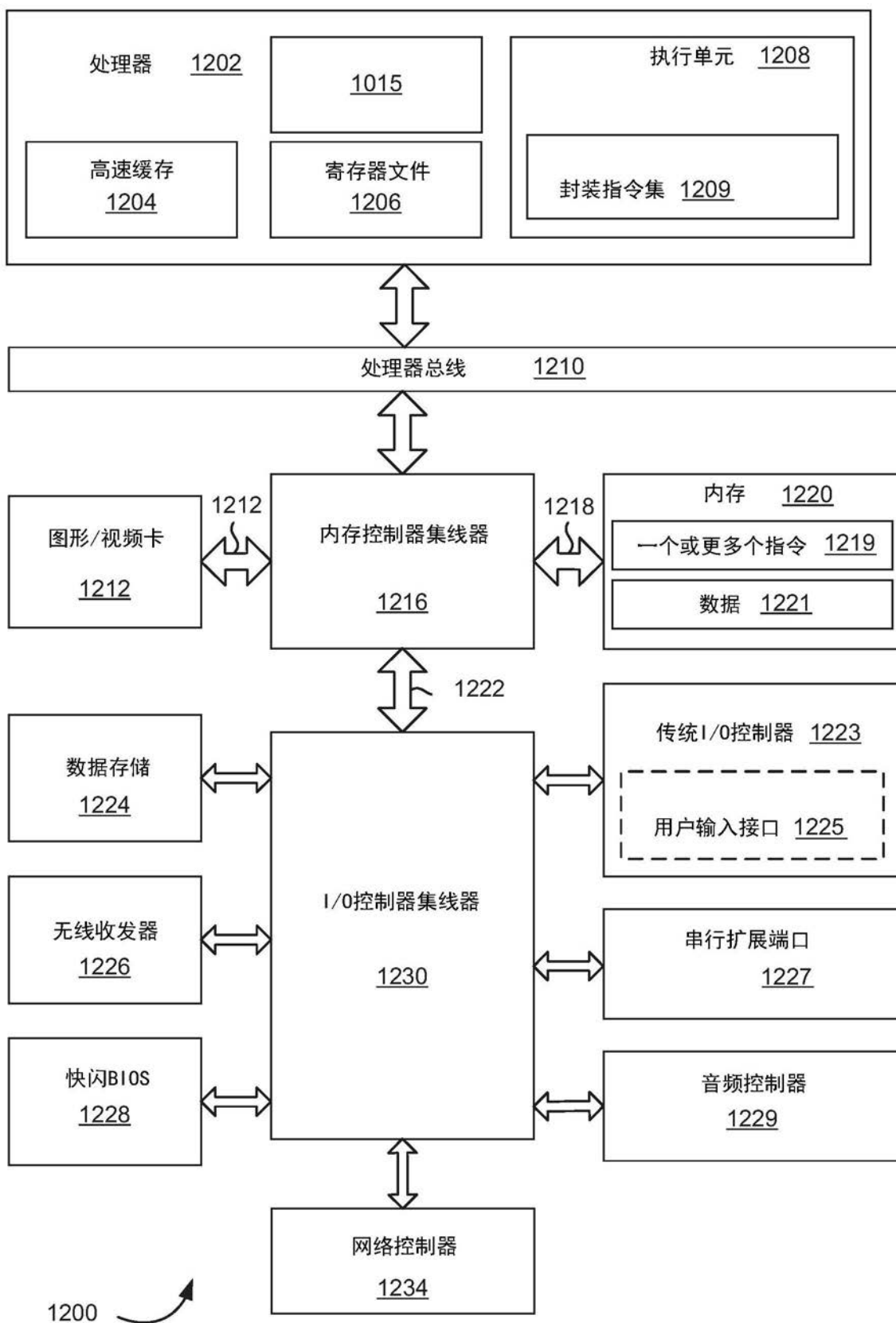


图12

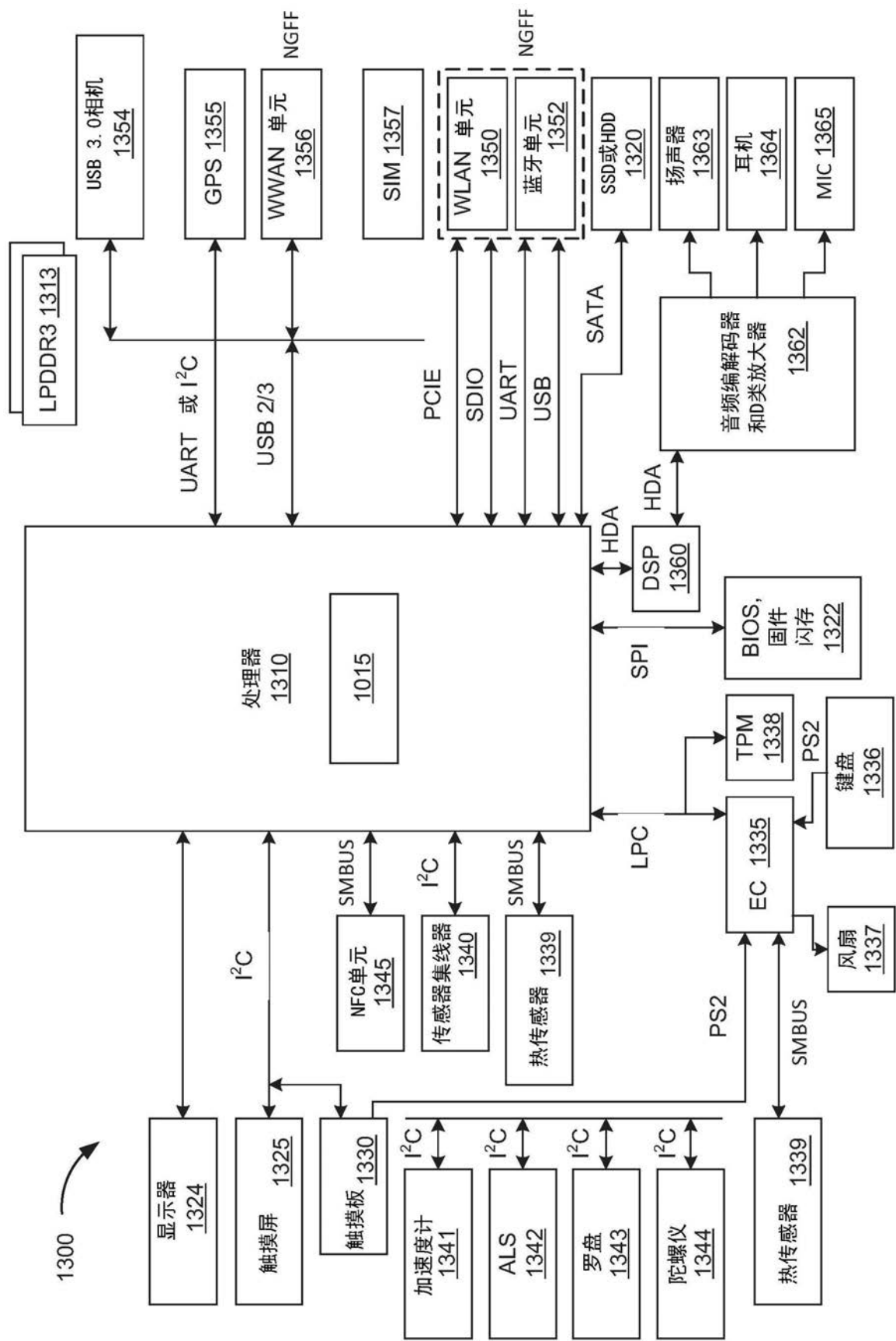


图13

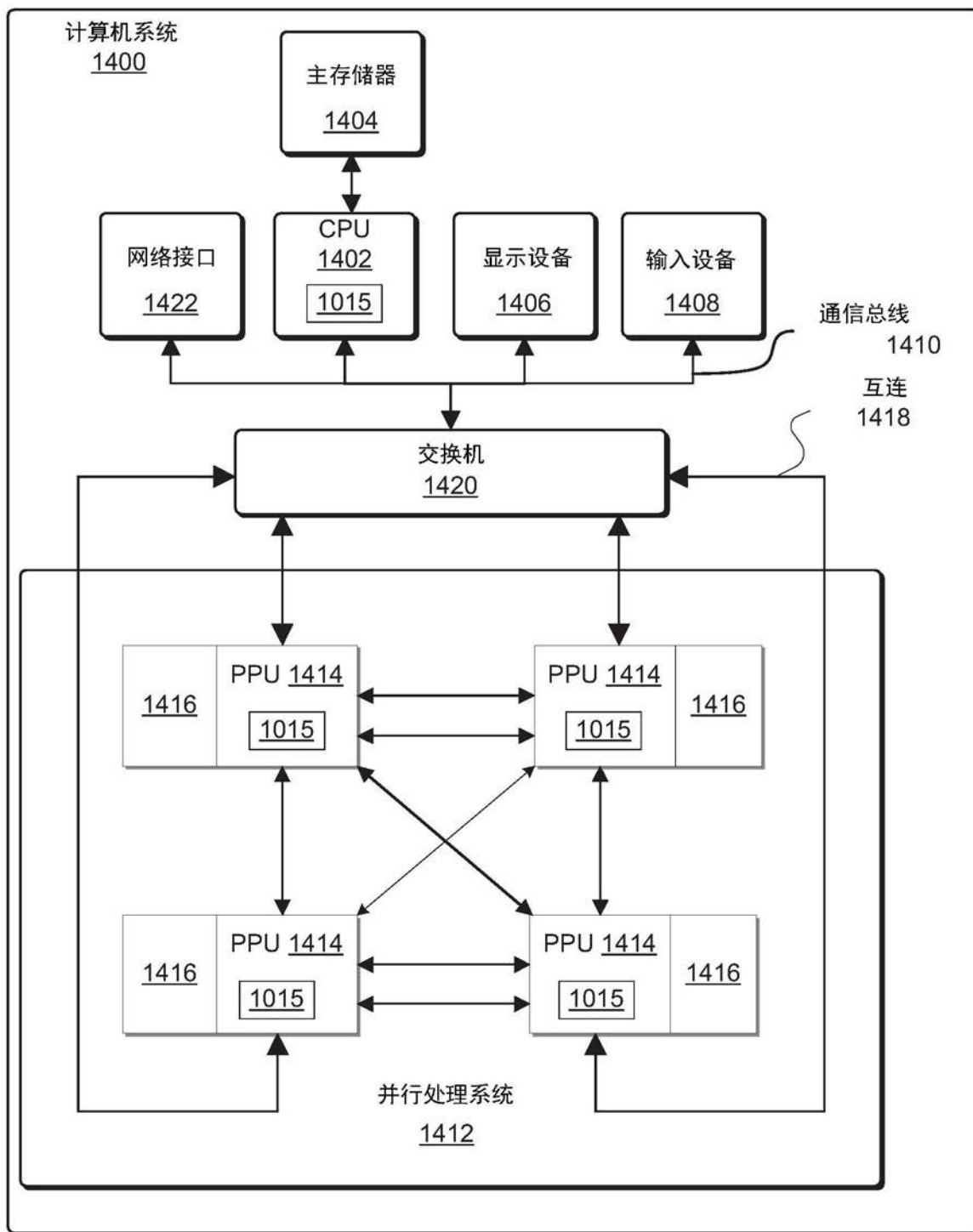


图14

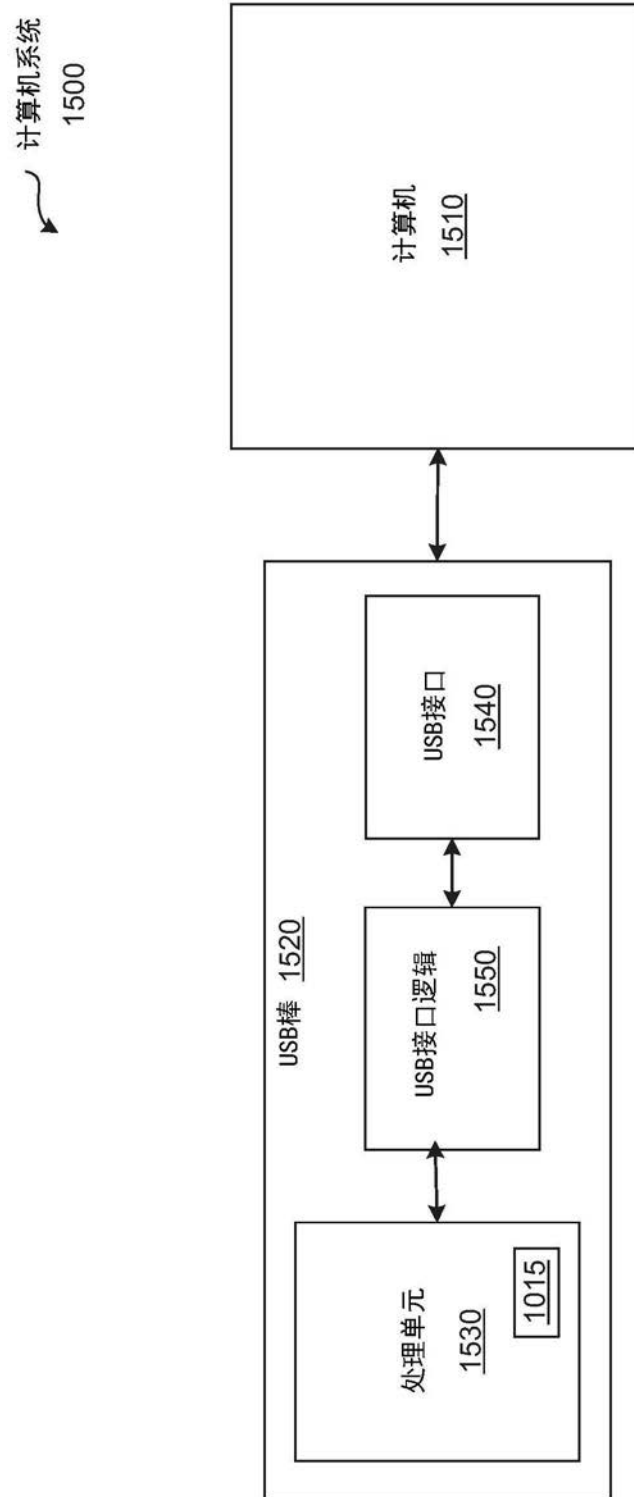


图15

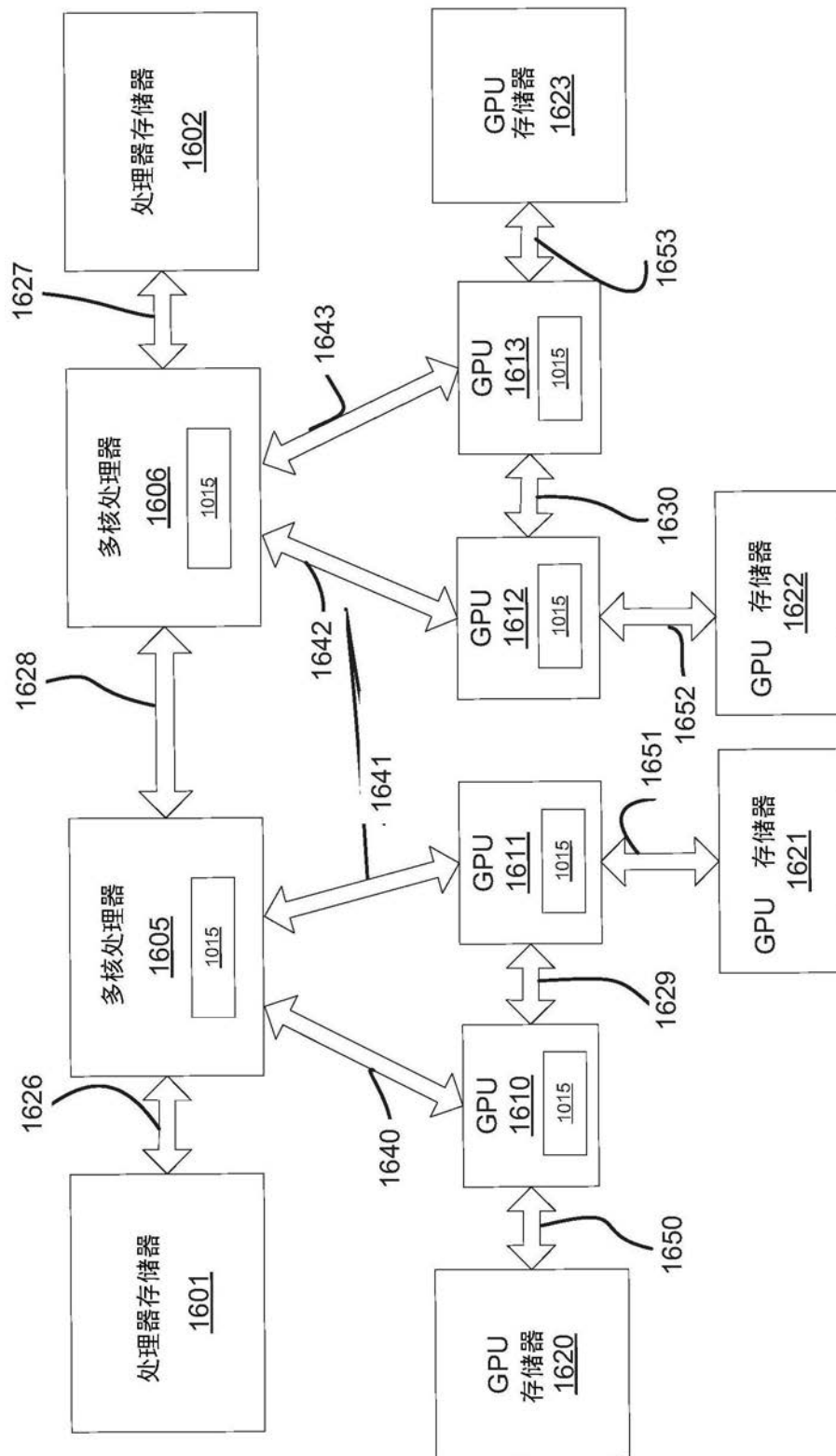


图16

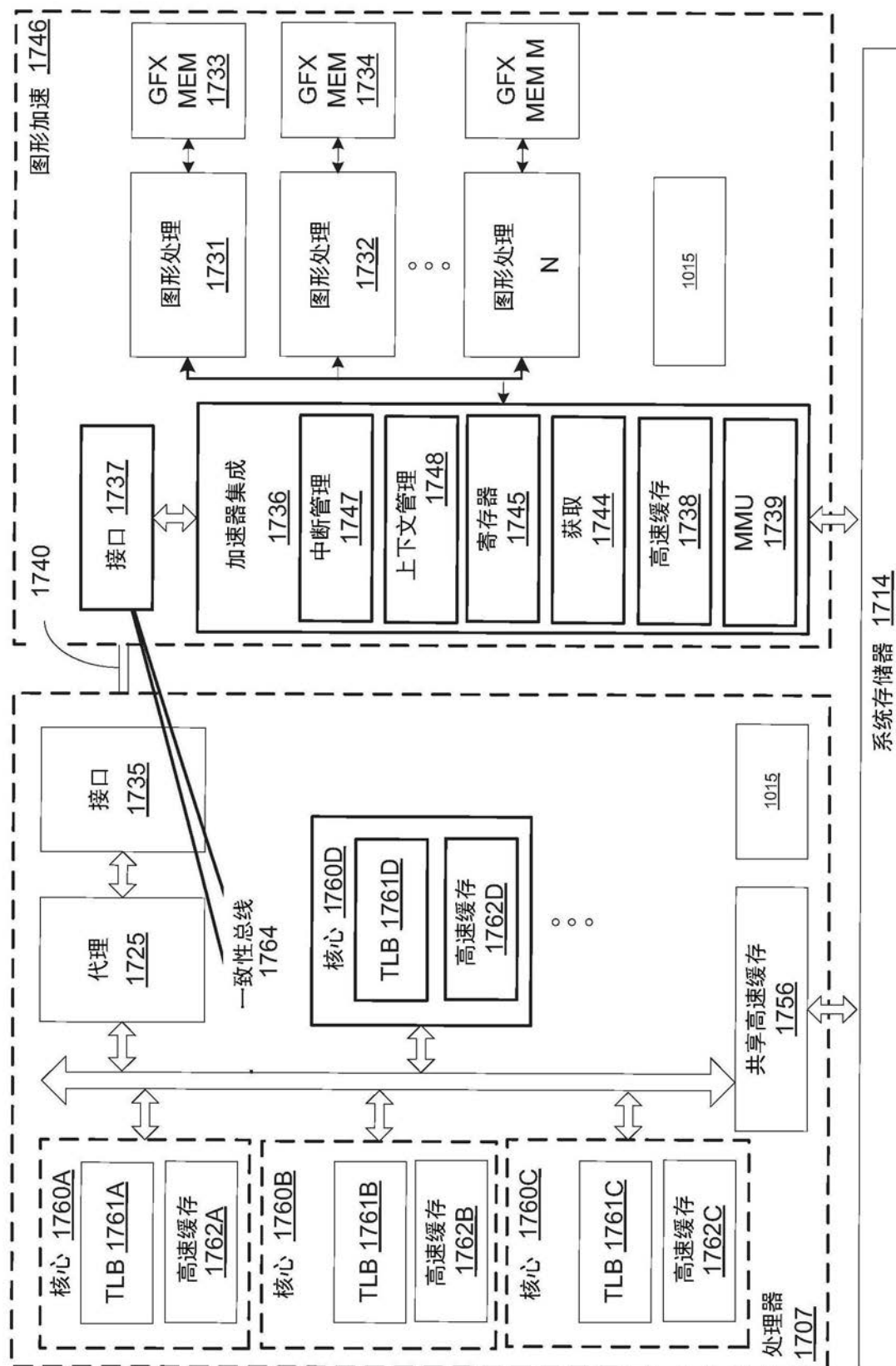


图17

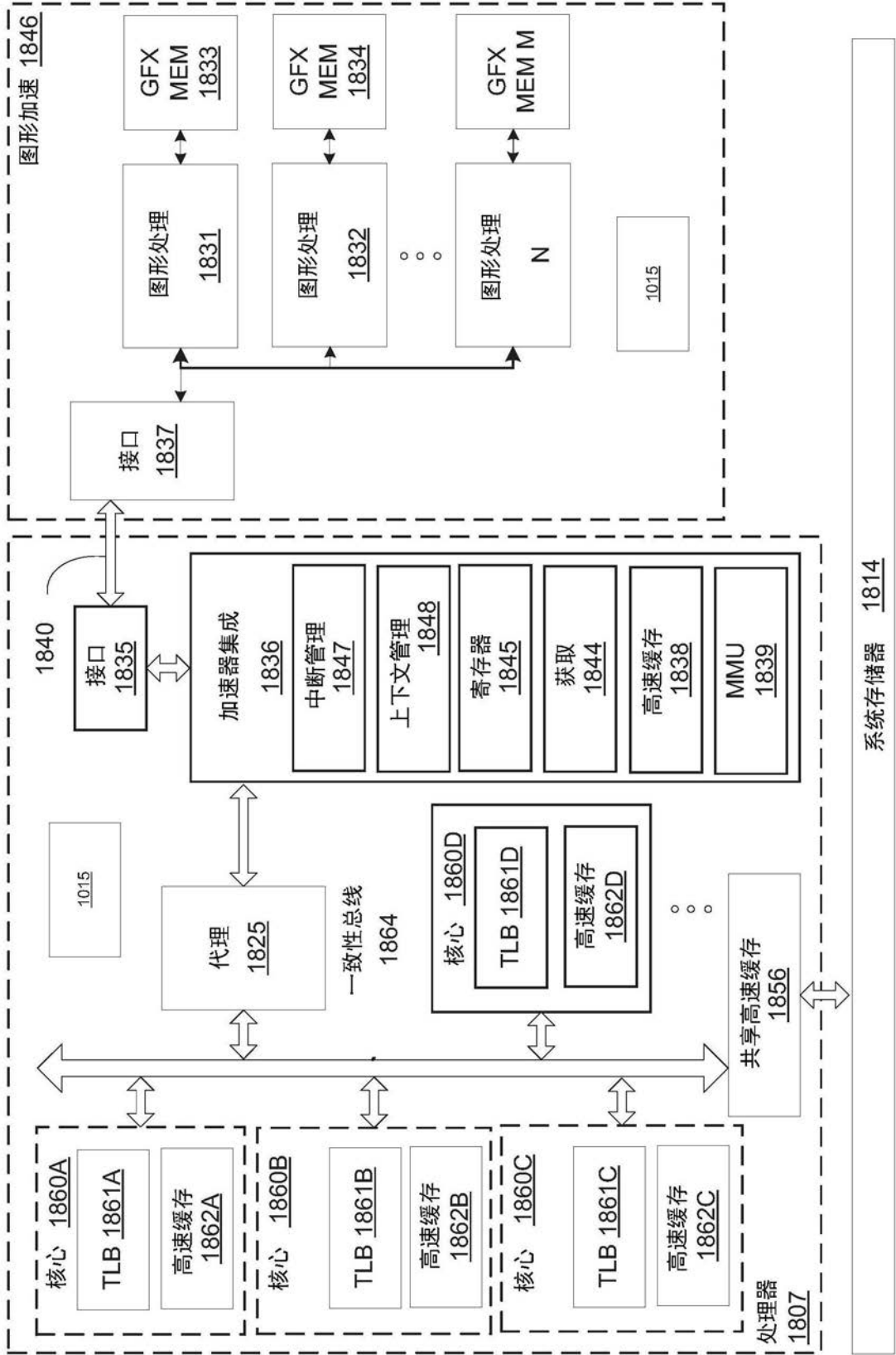


图18

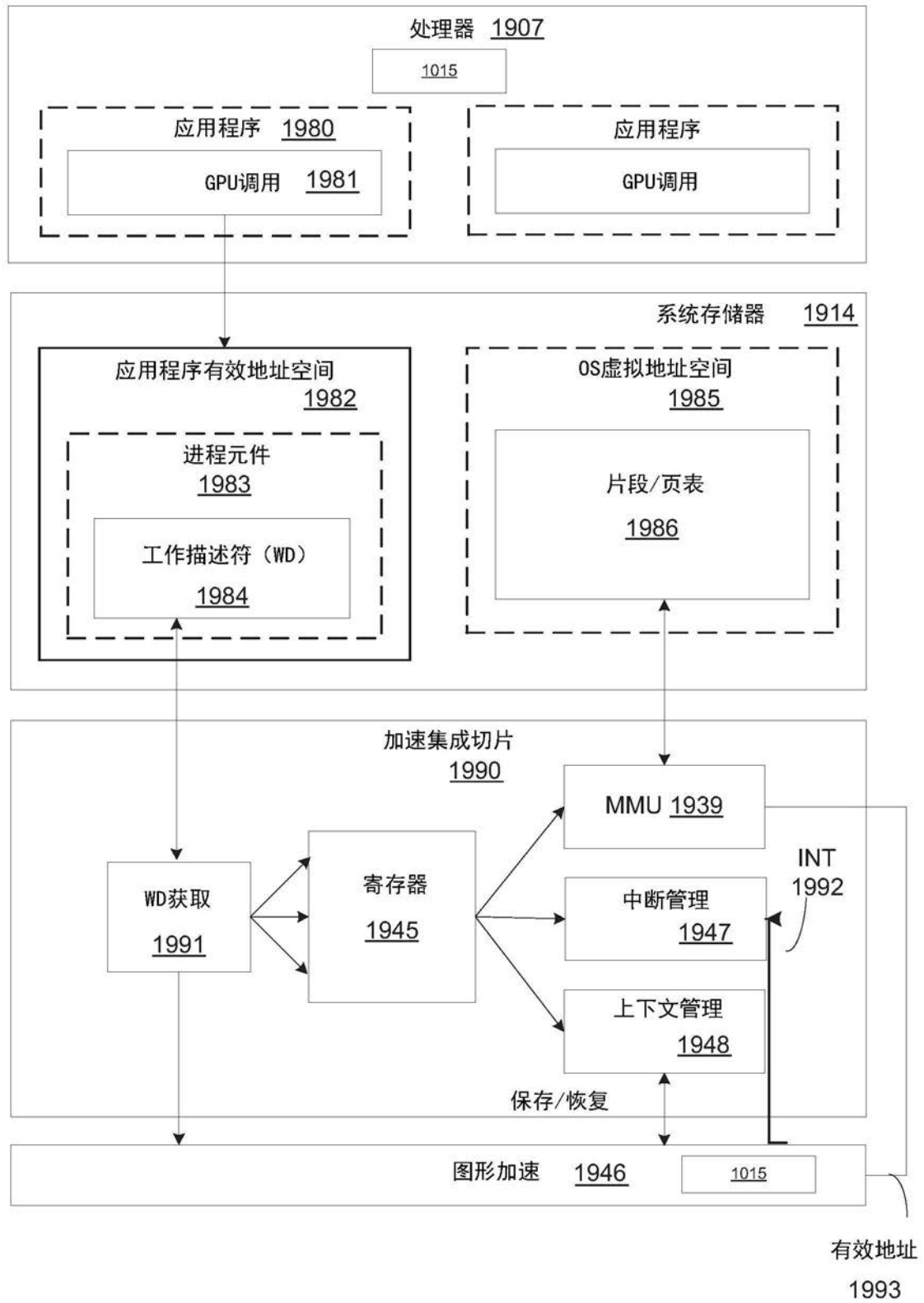


图19

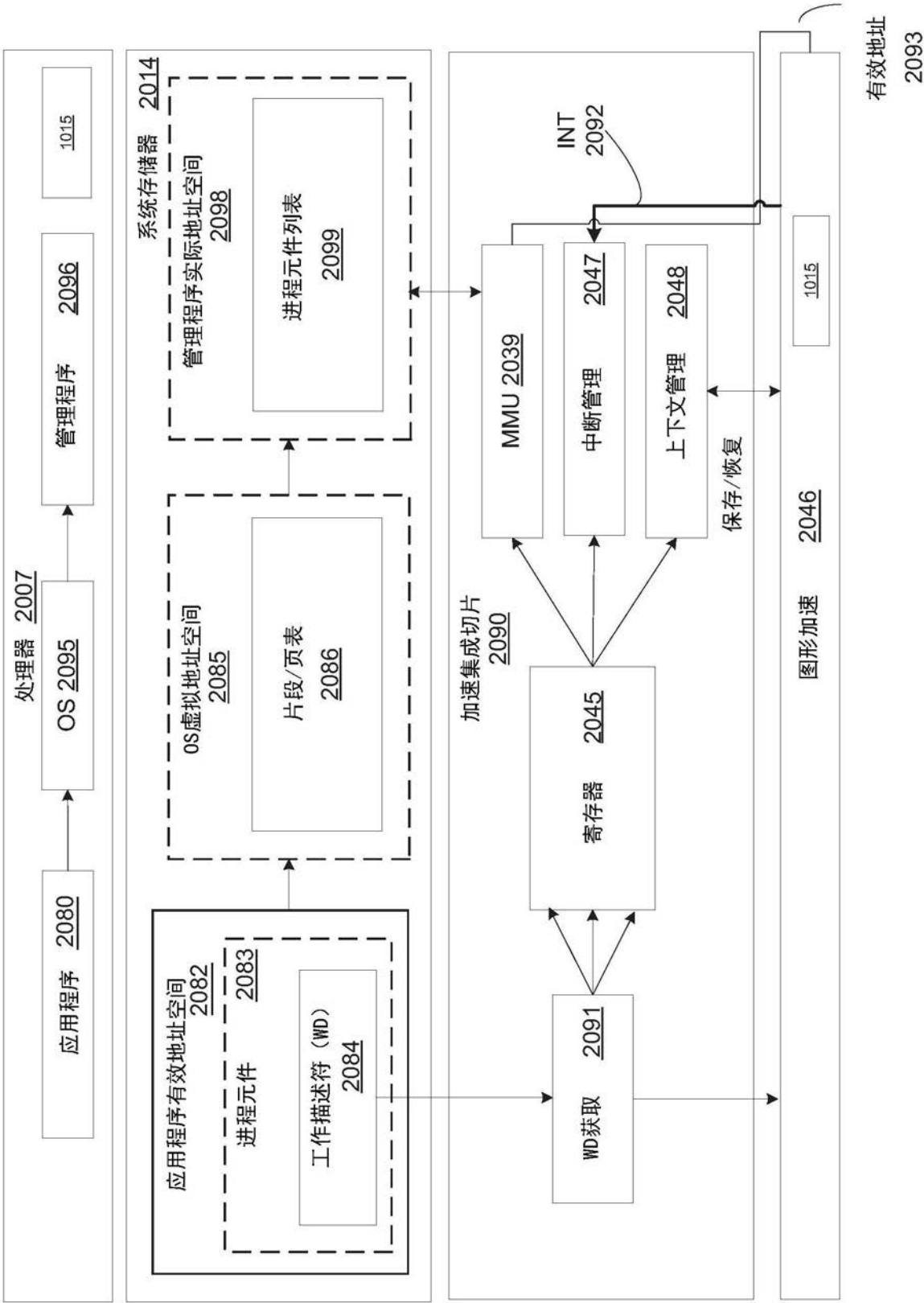


图20

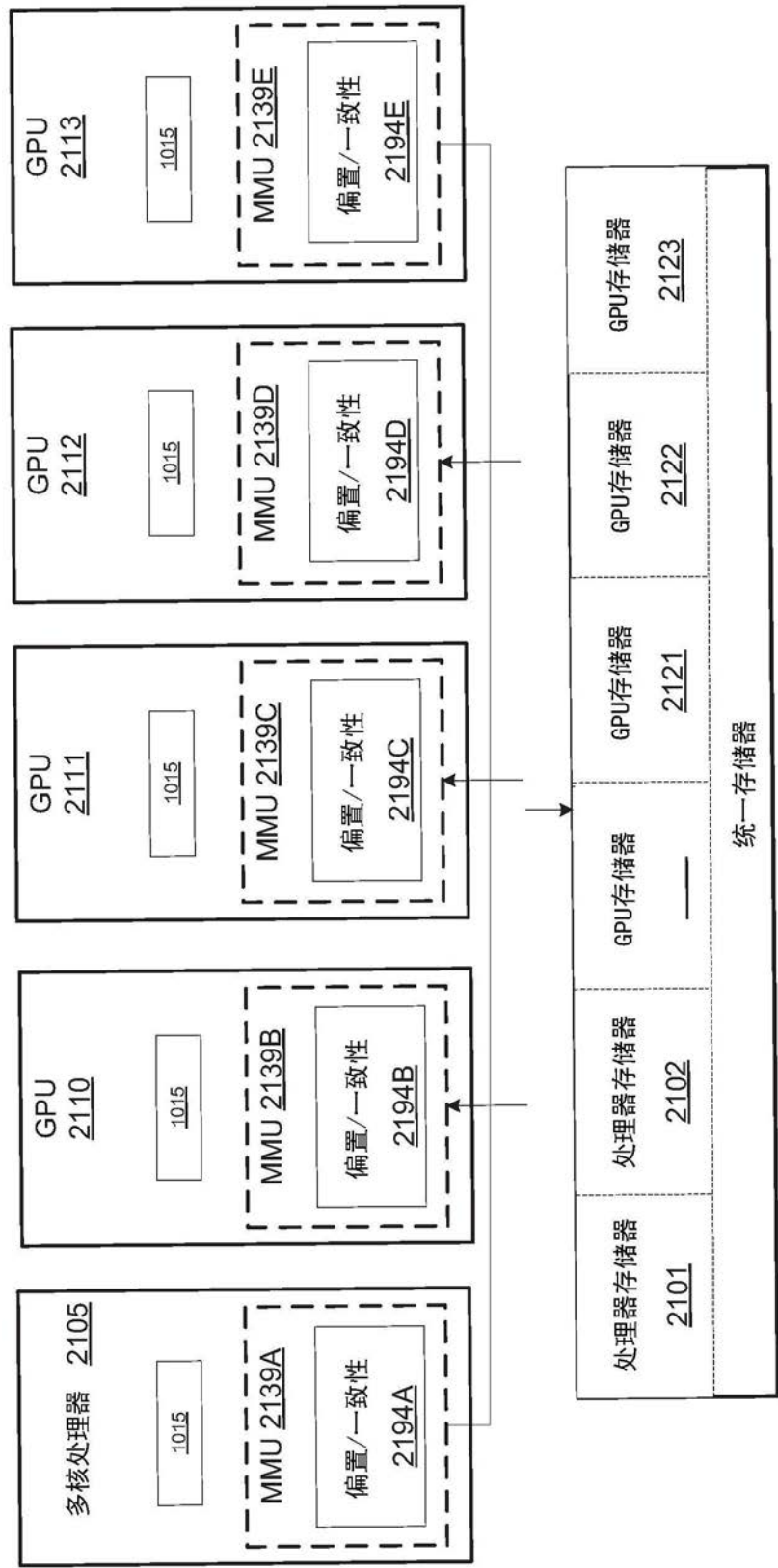


图21

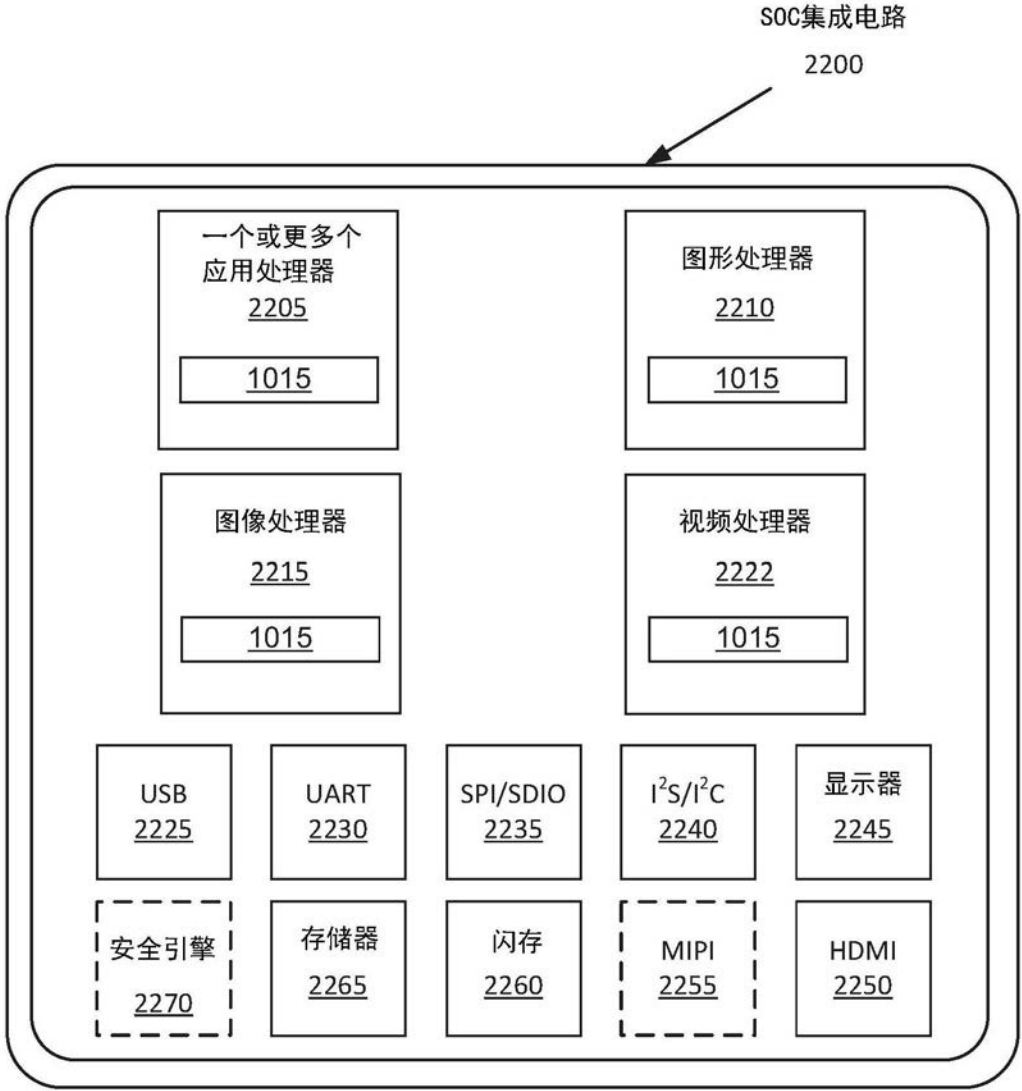


图22

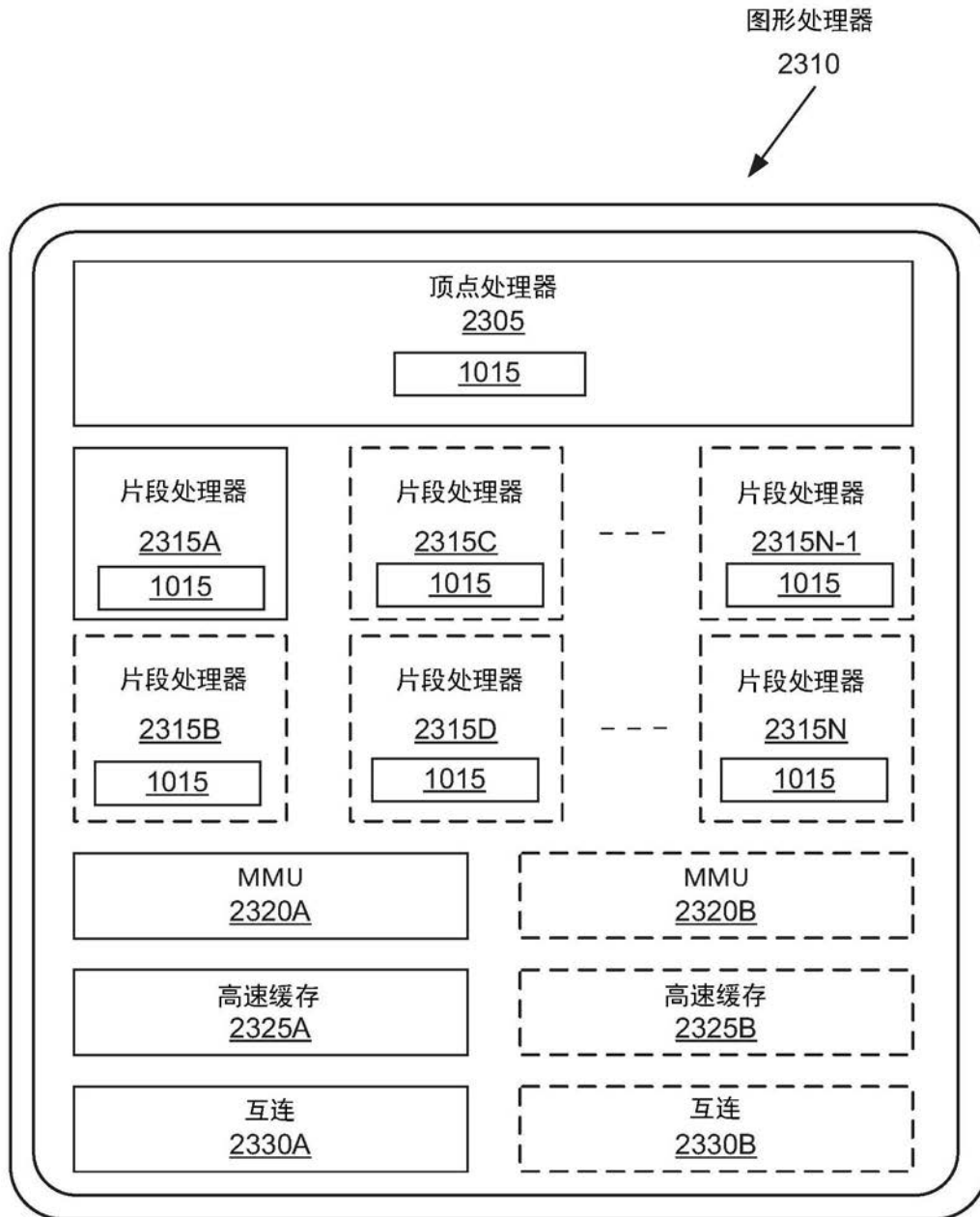


图23

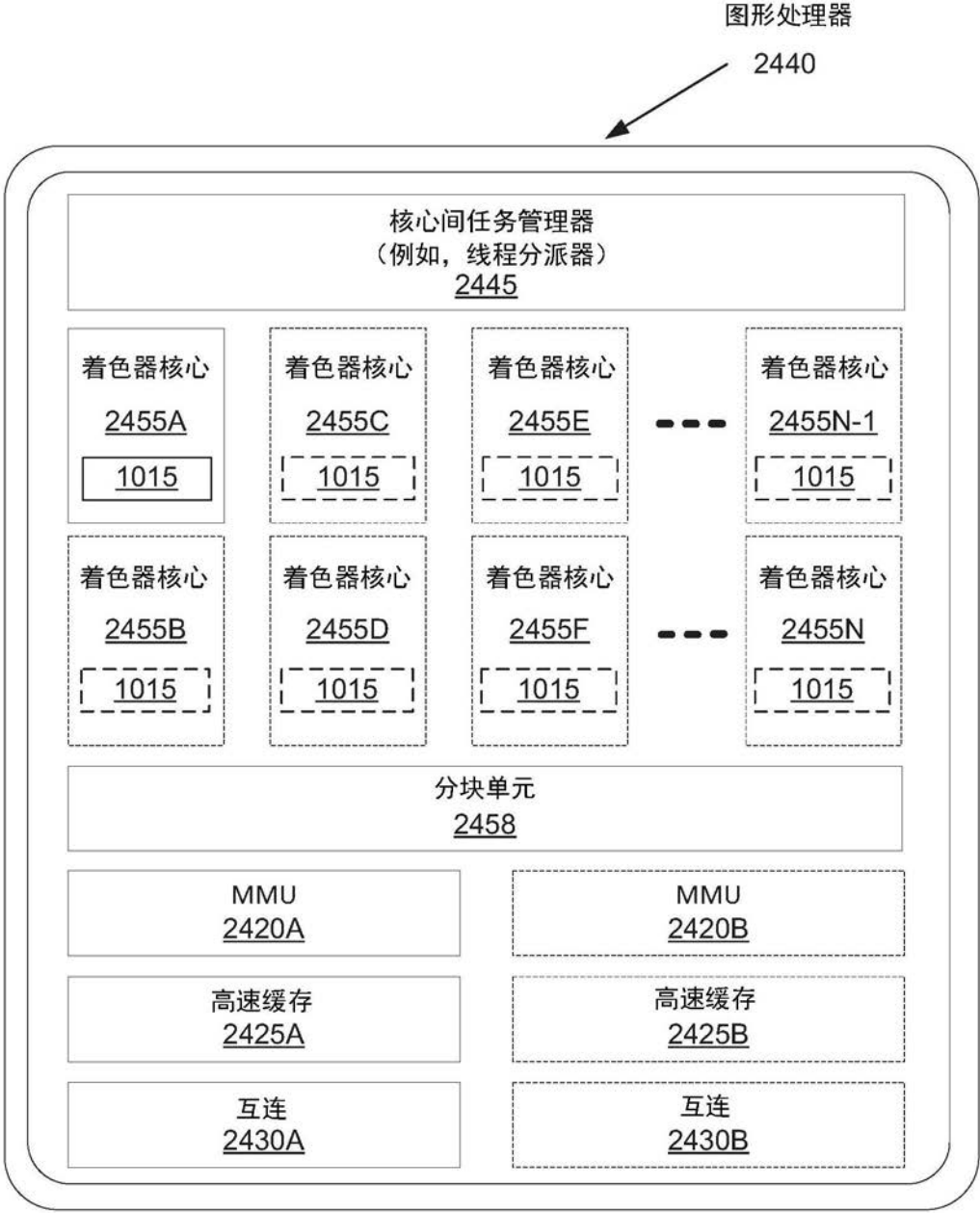


图24

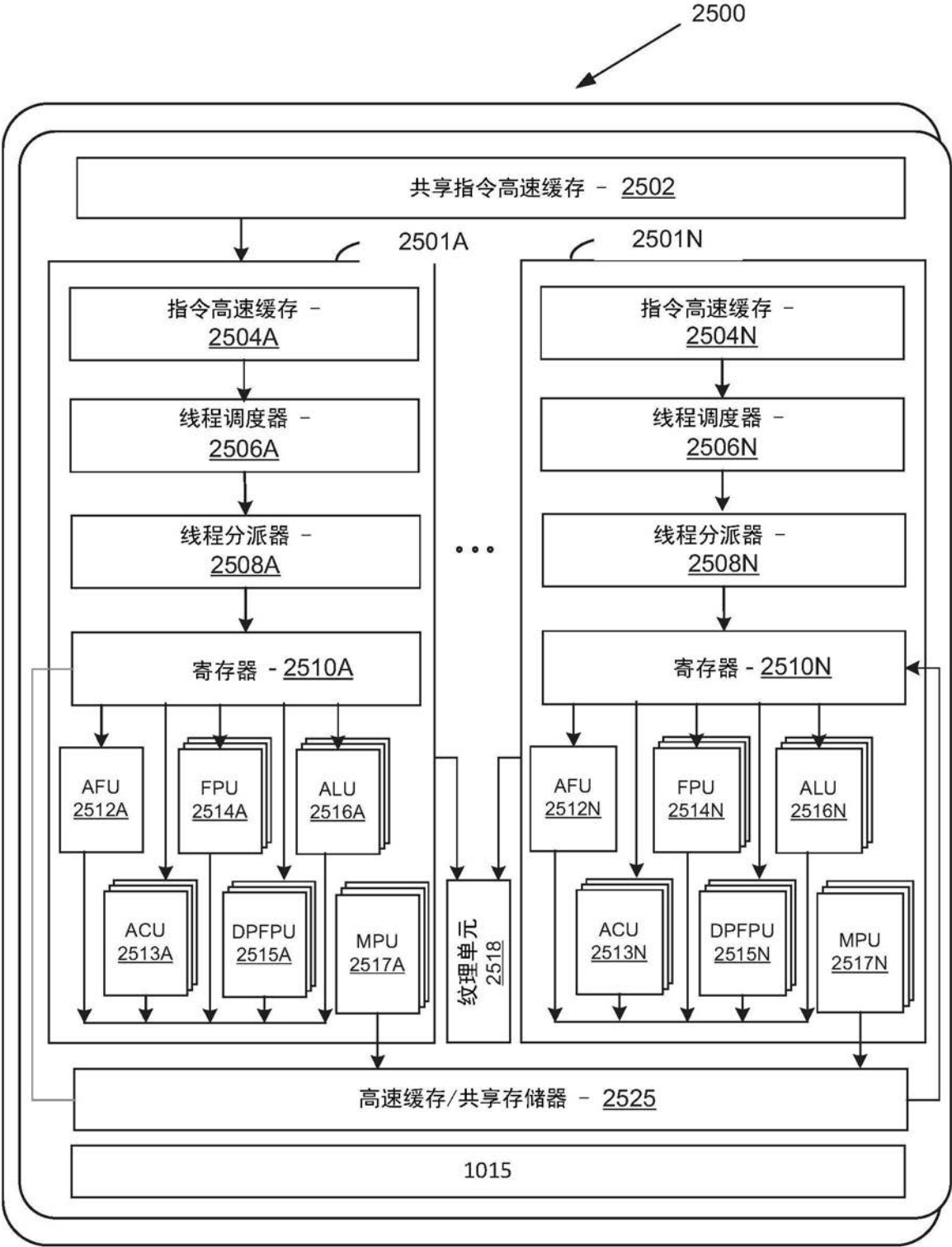


图25

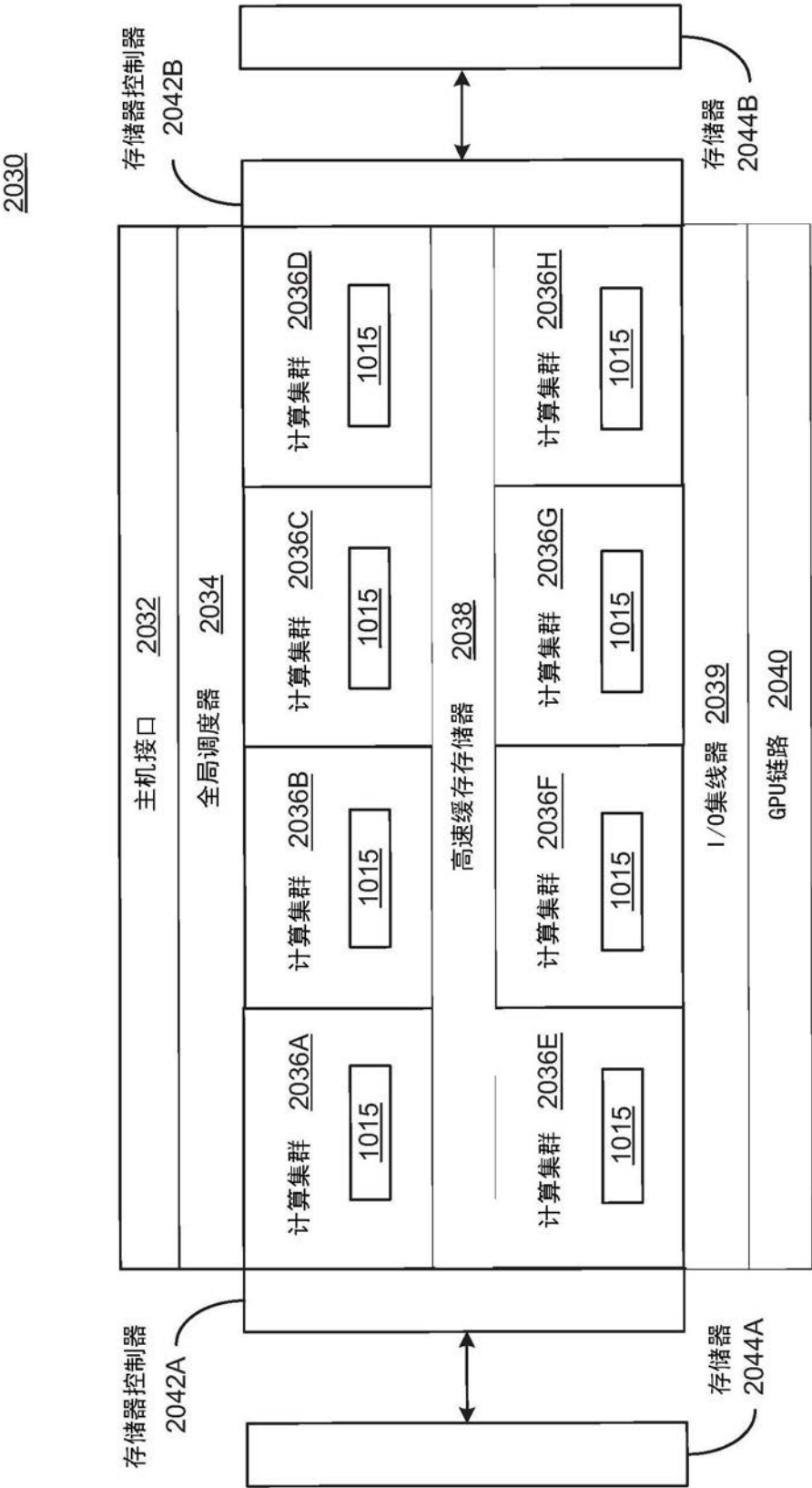


图26

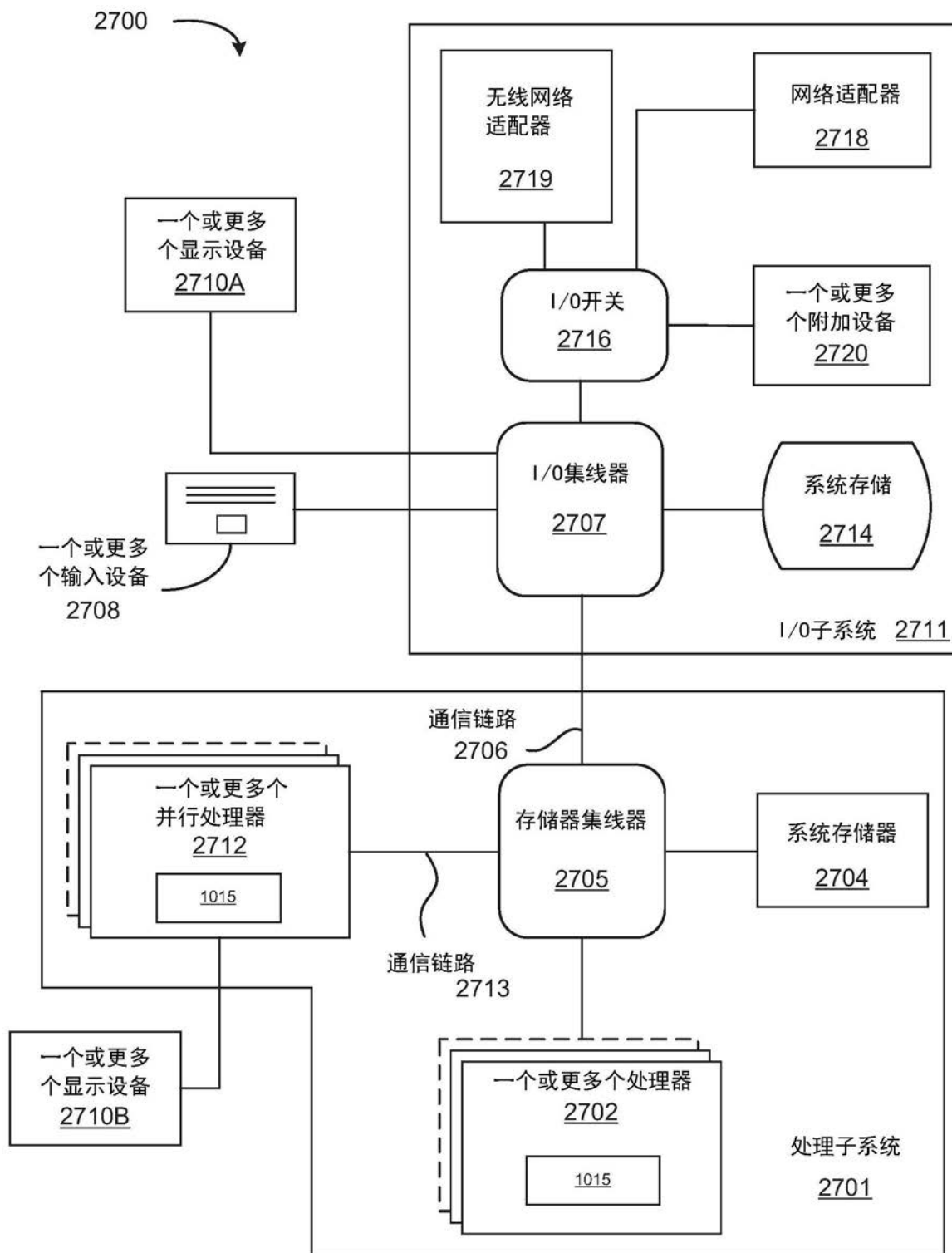


图27

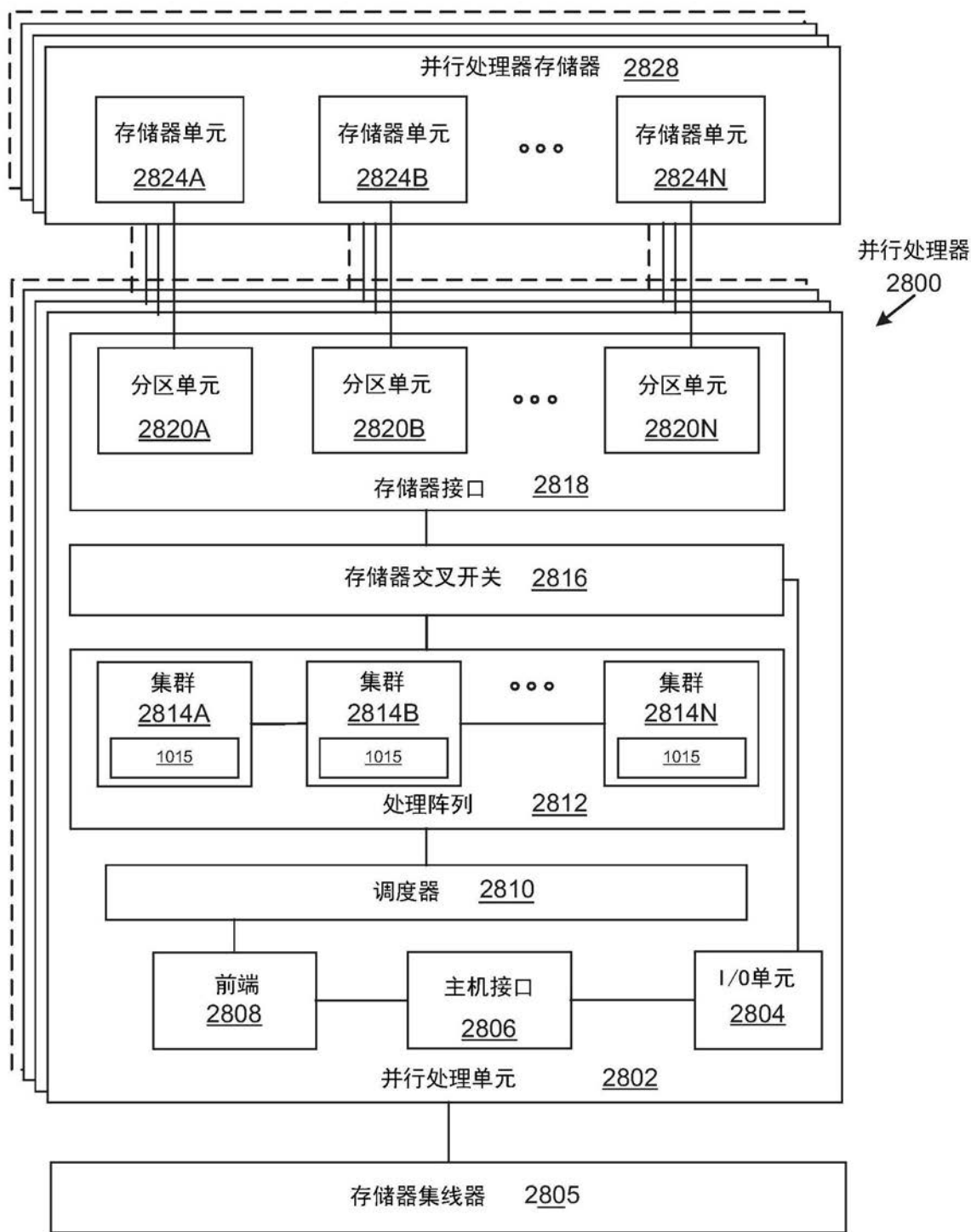


图28

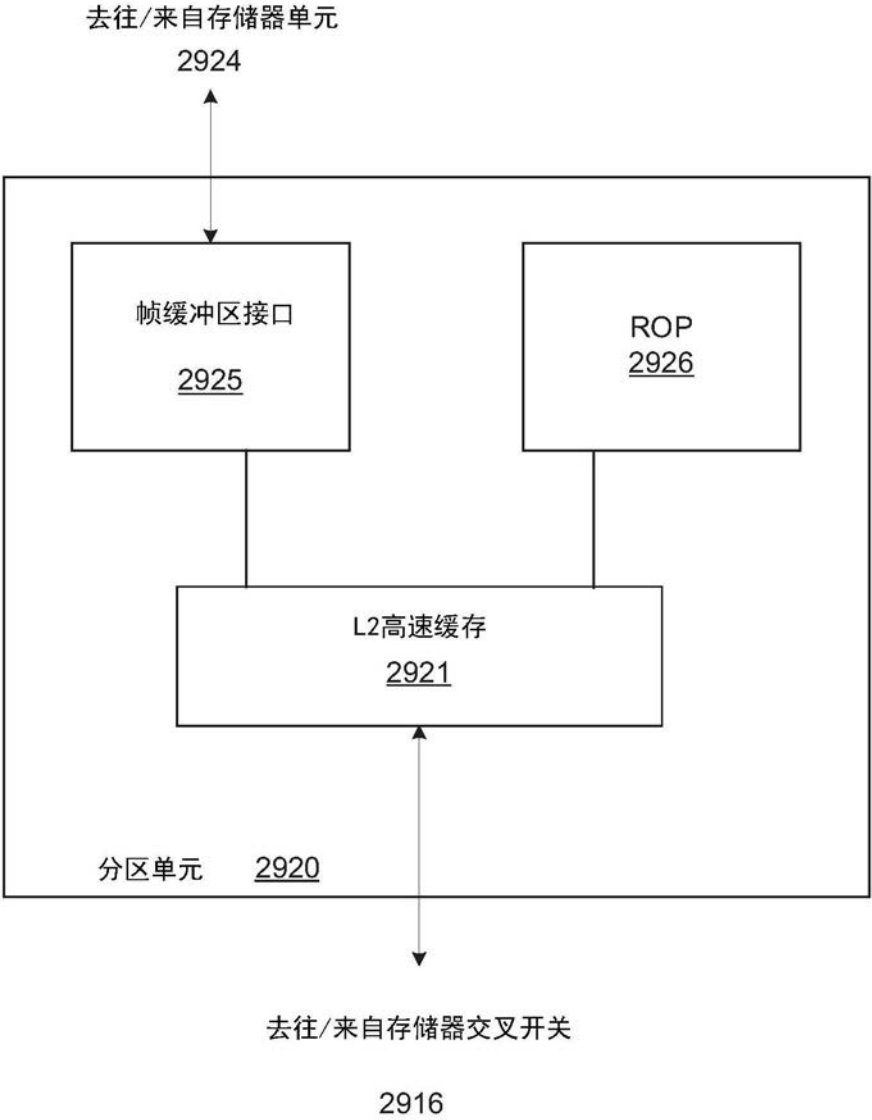


图29

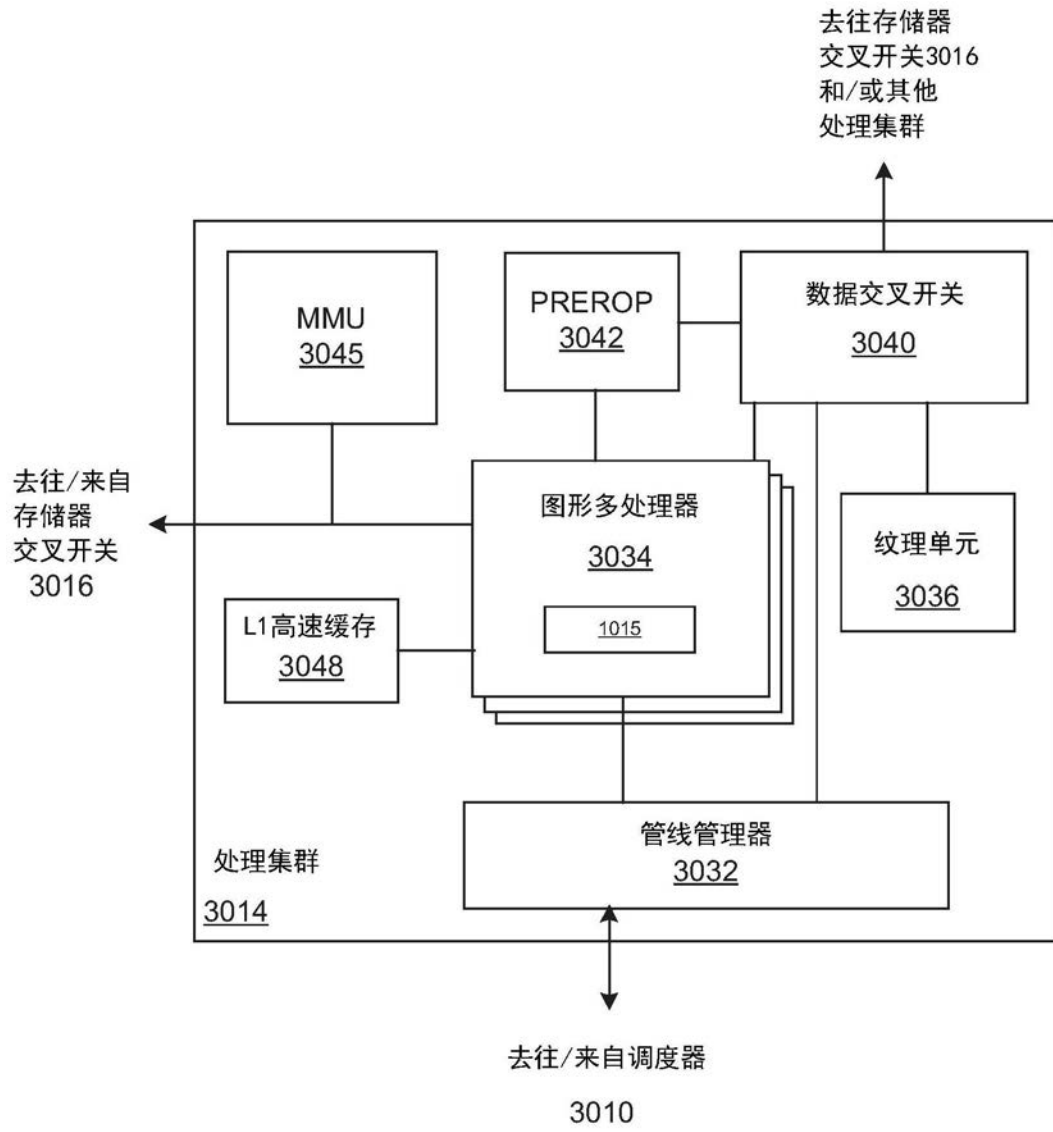


图30

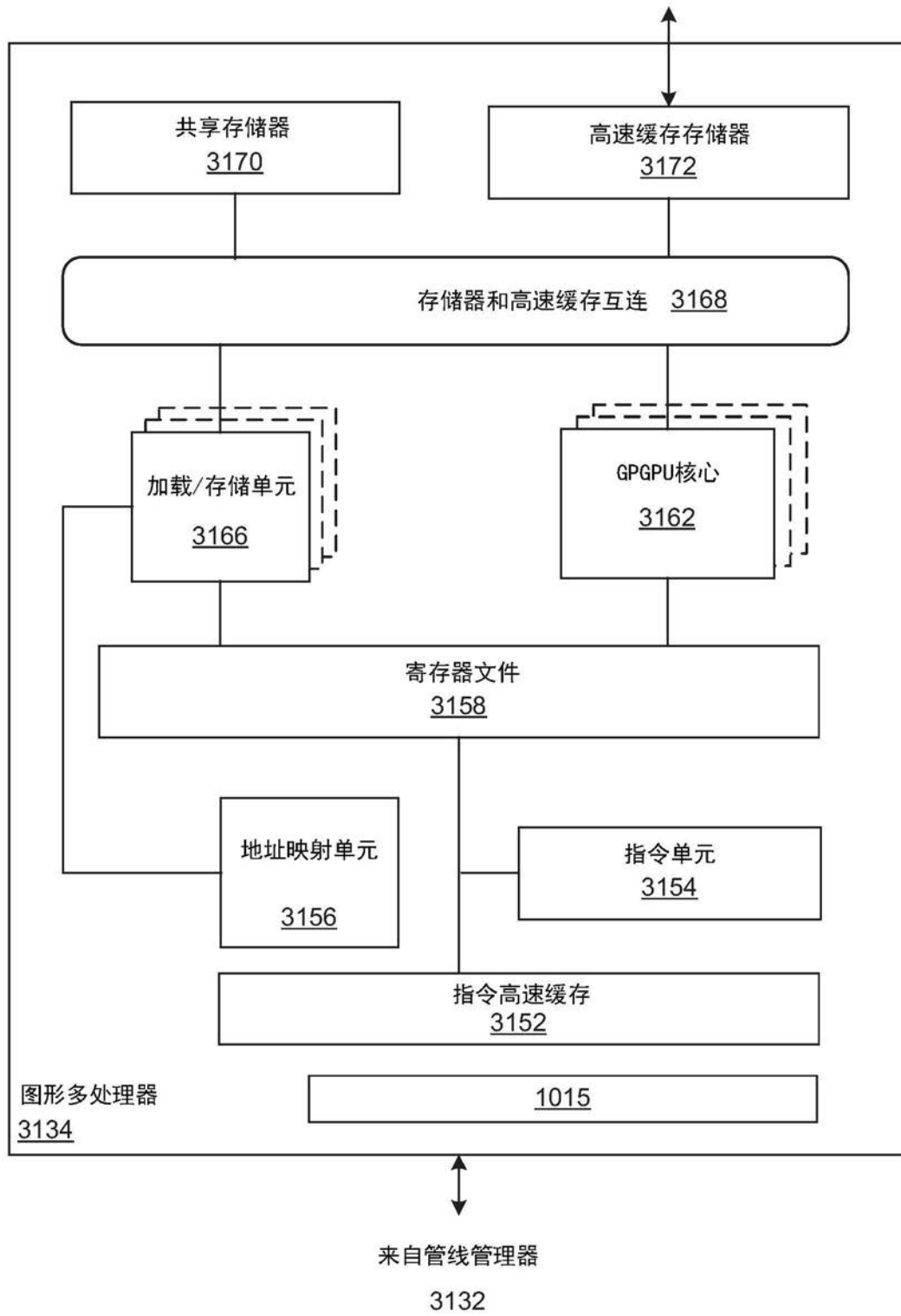


图31

3200

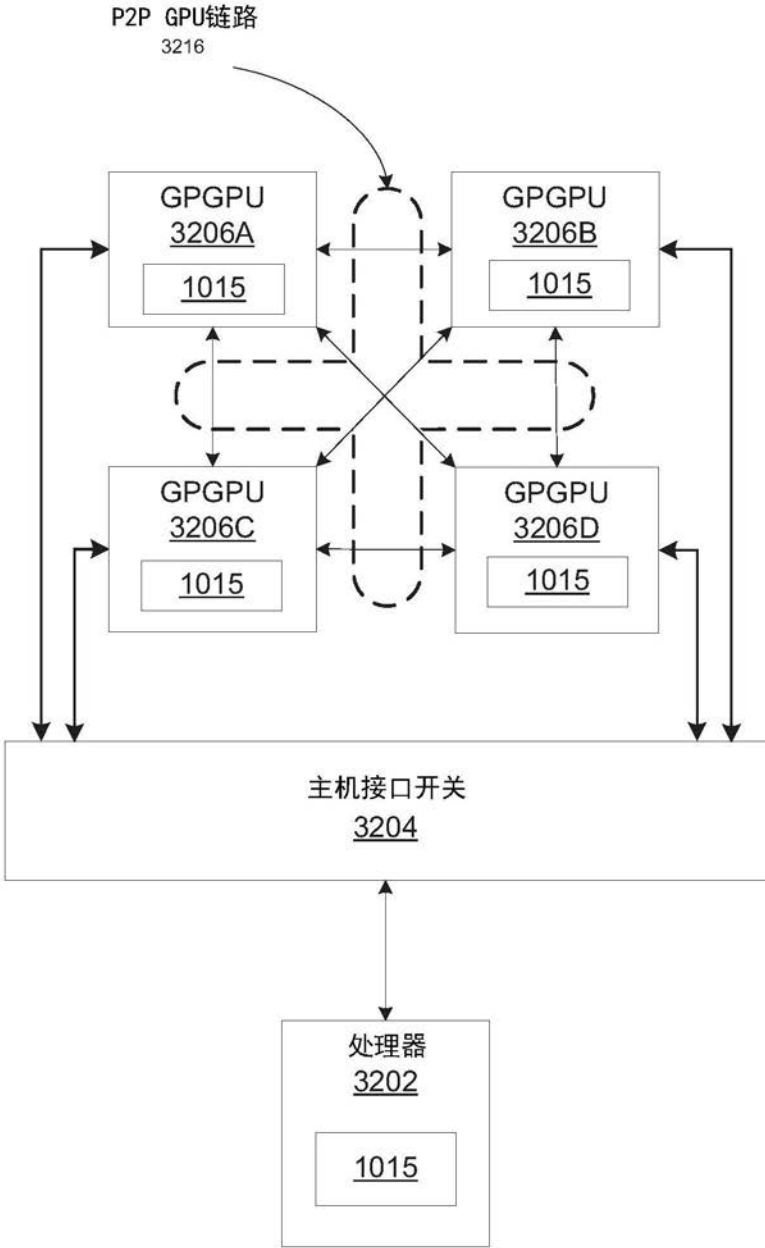


图32

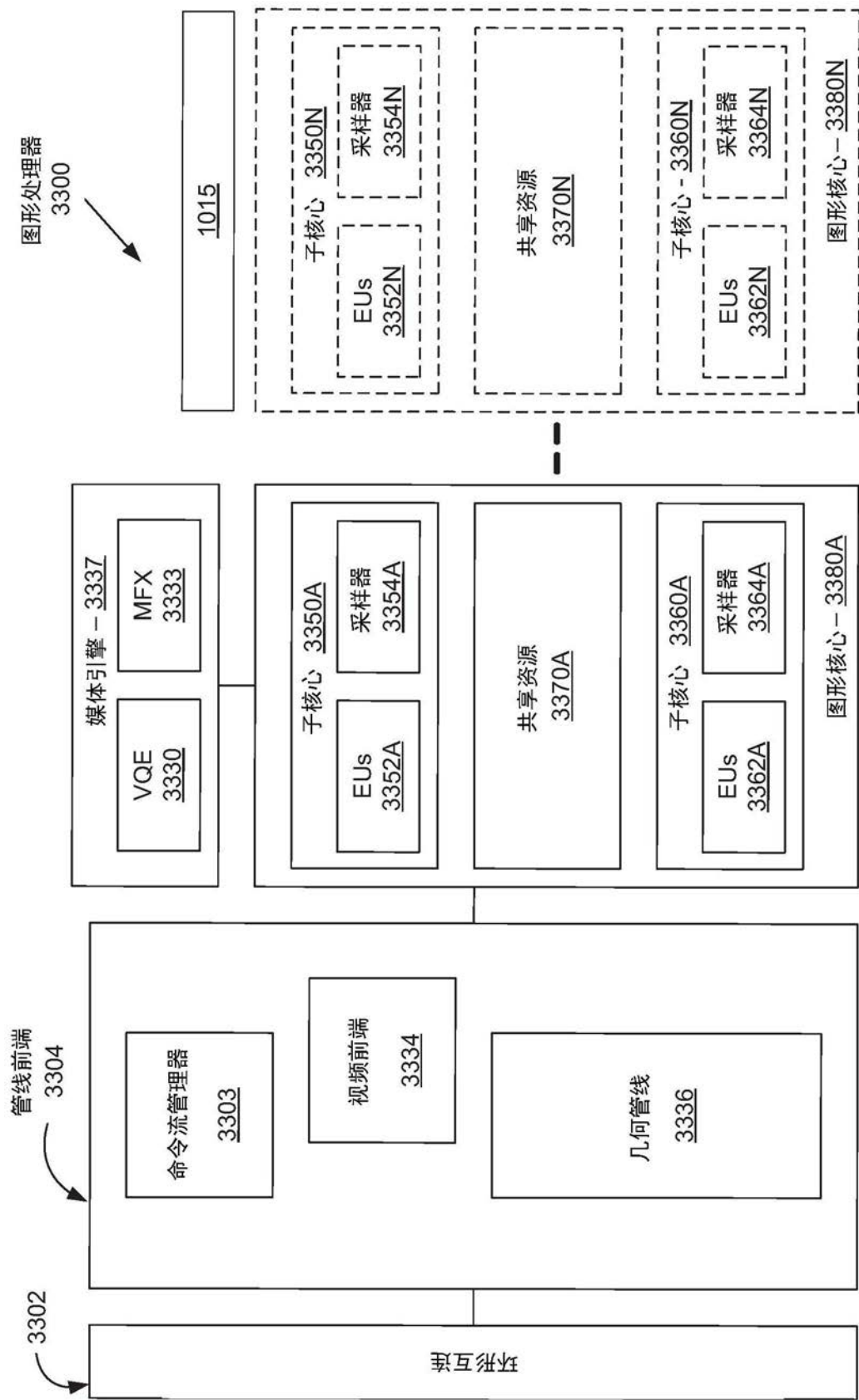


图33

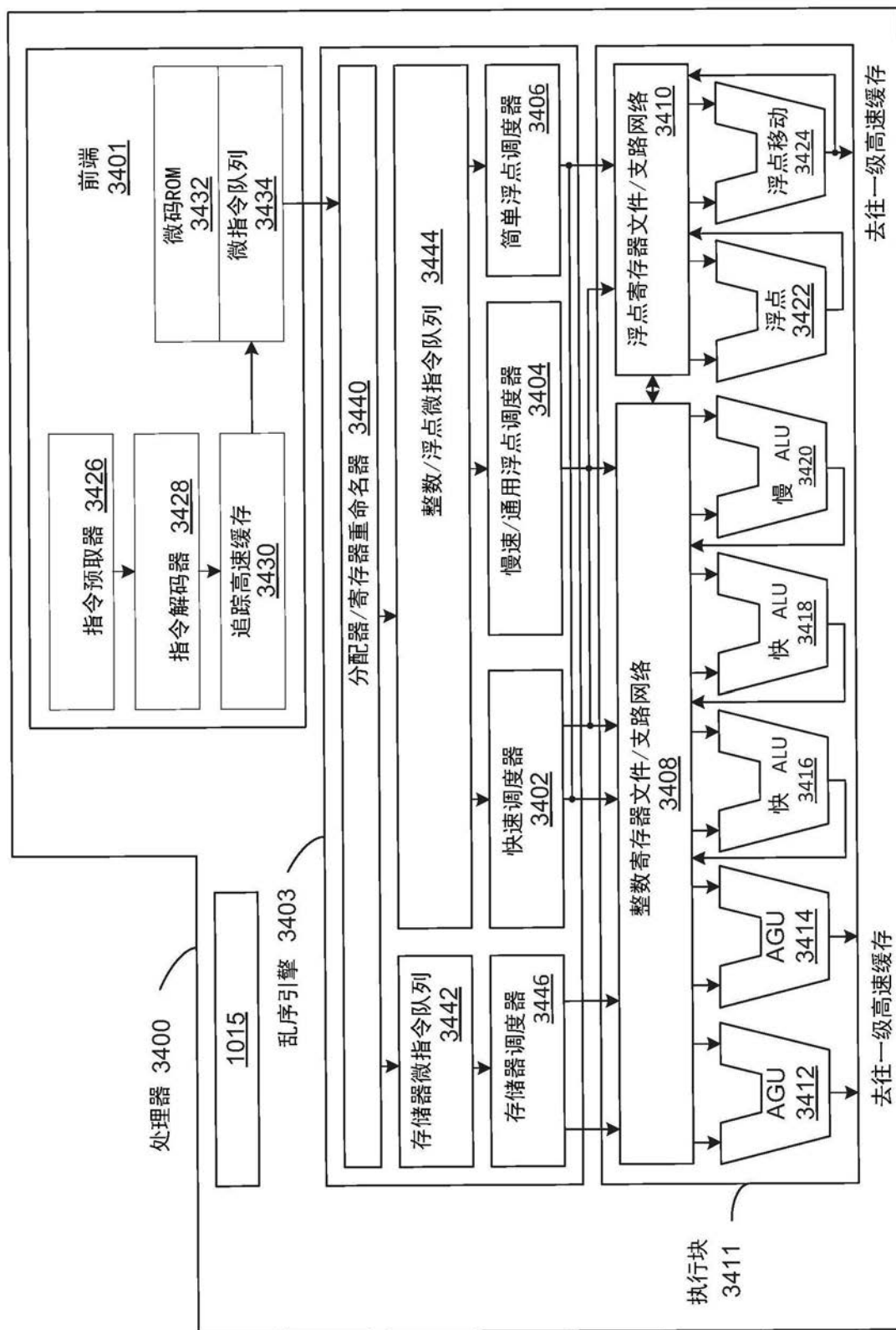


图34

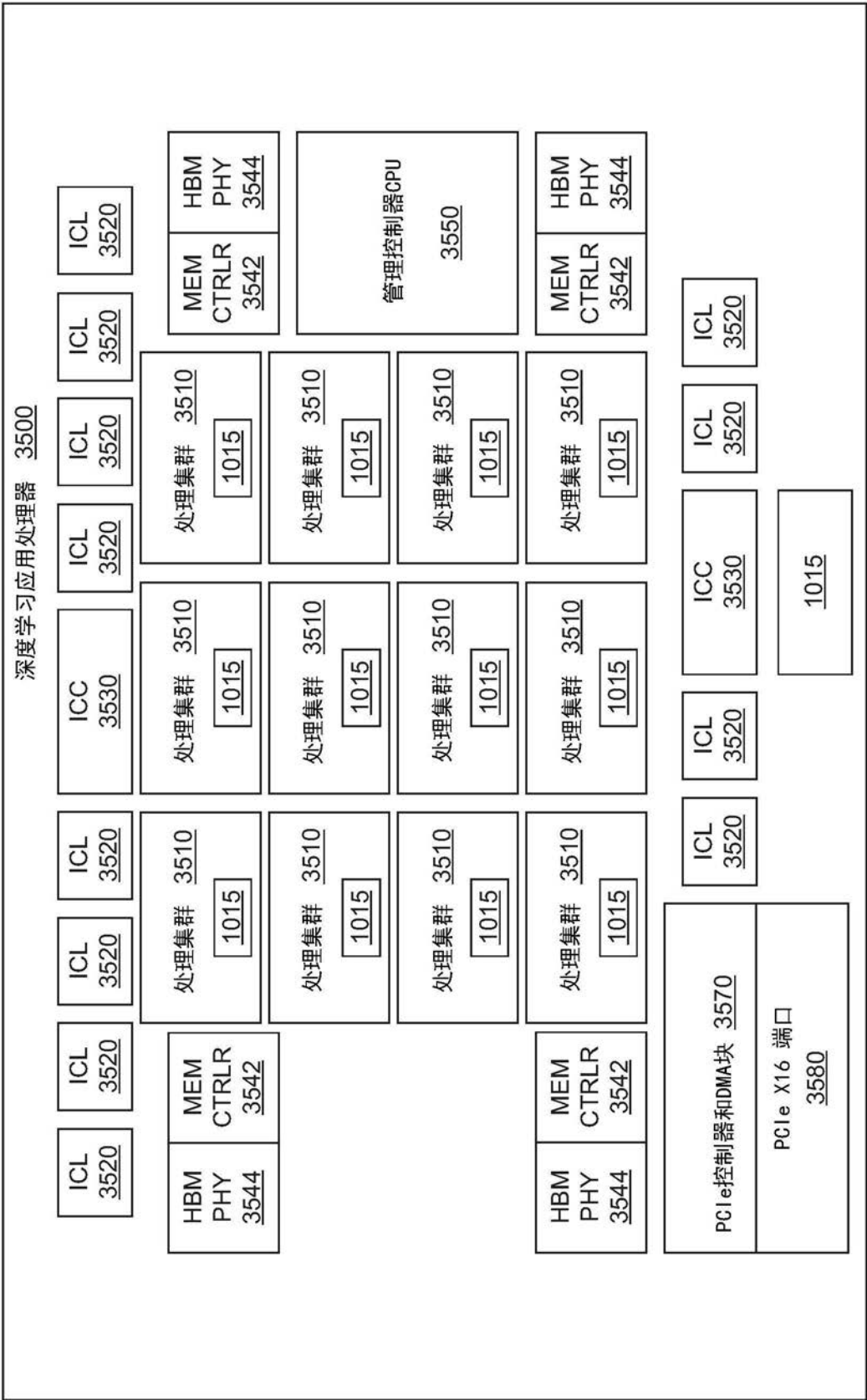


图35

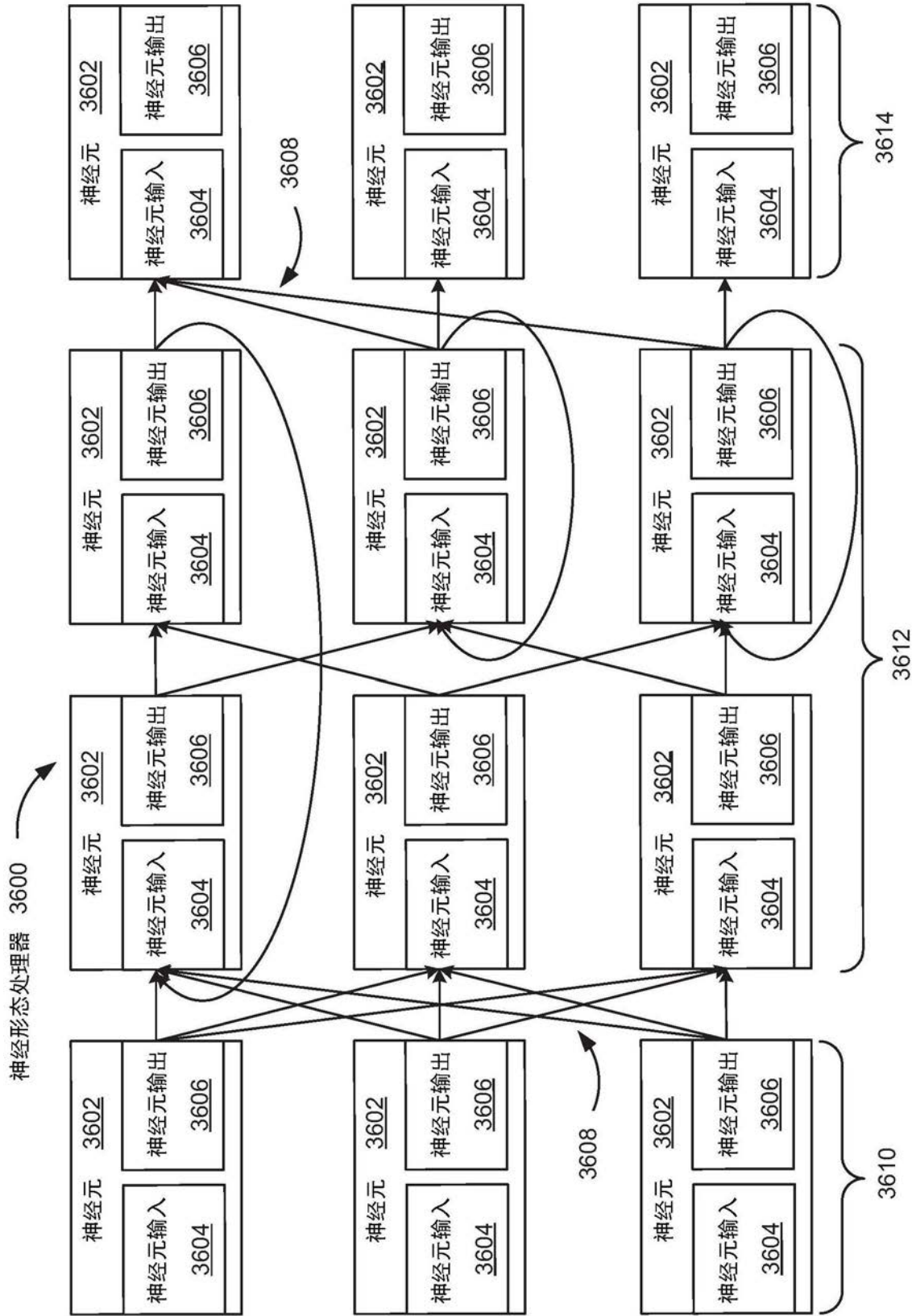


图36

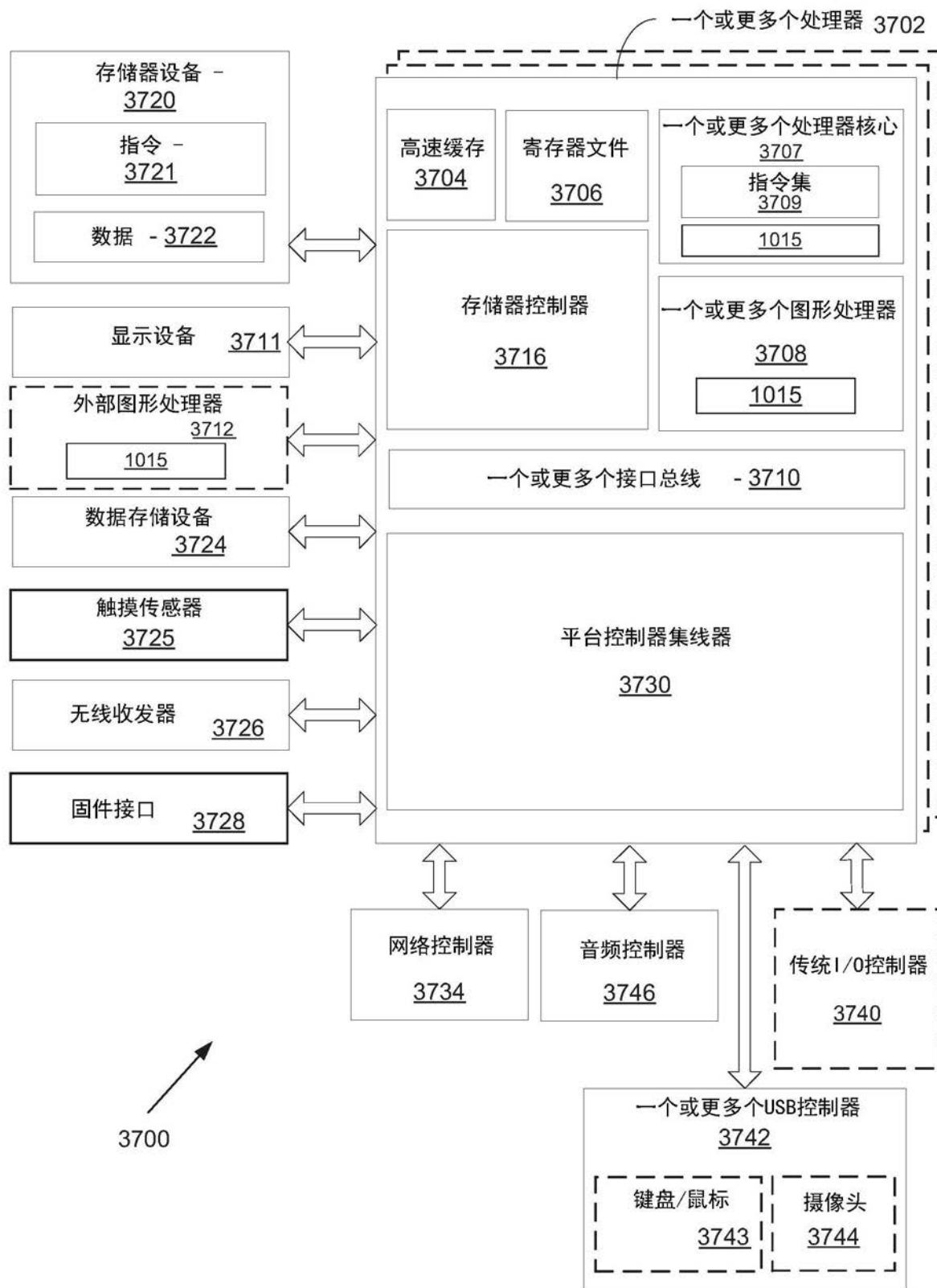


图37

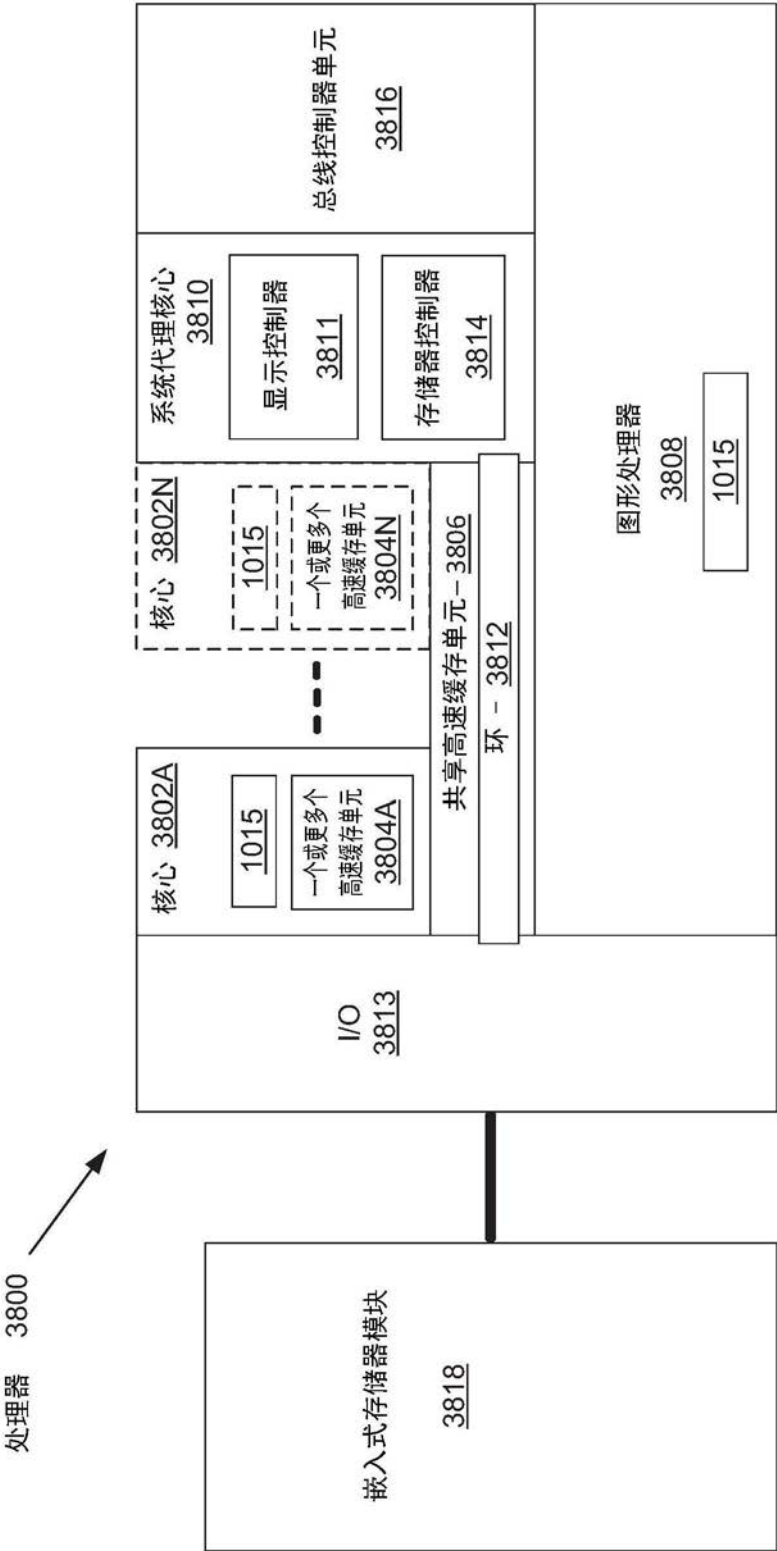


图38

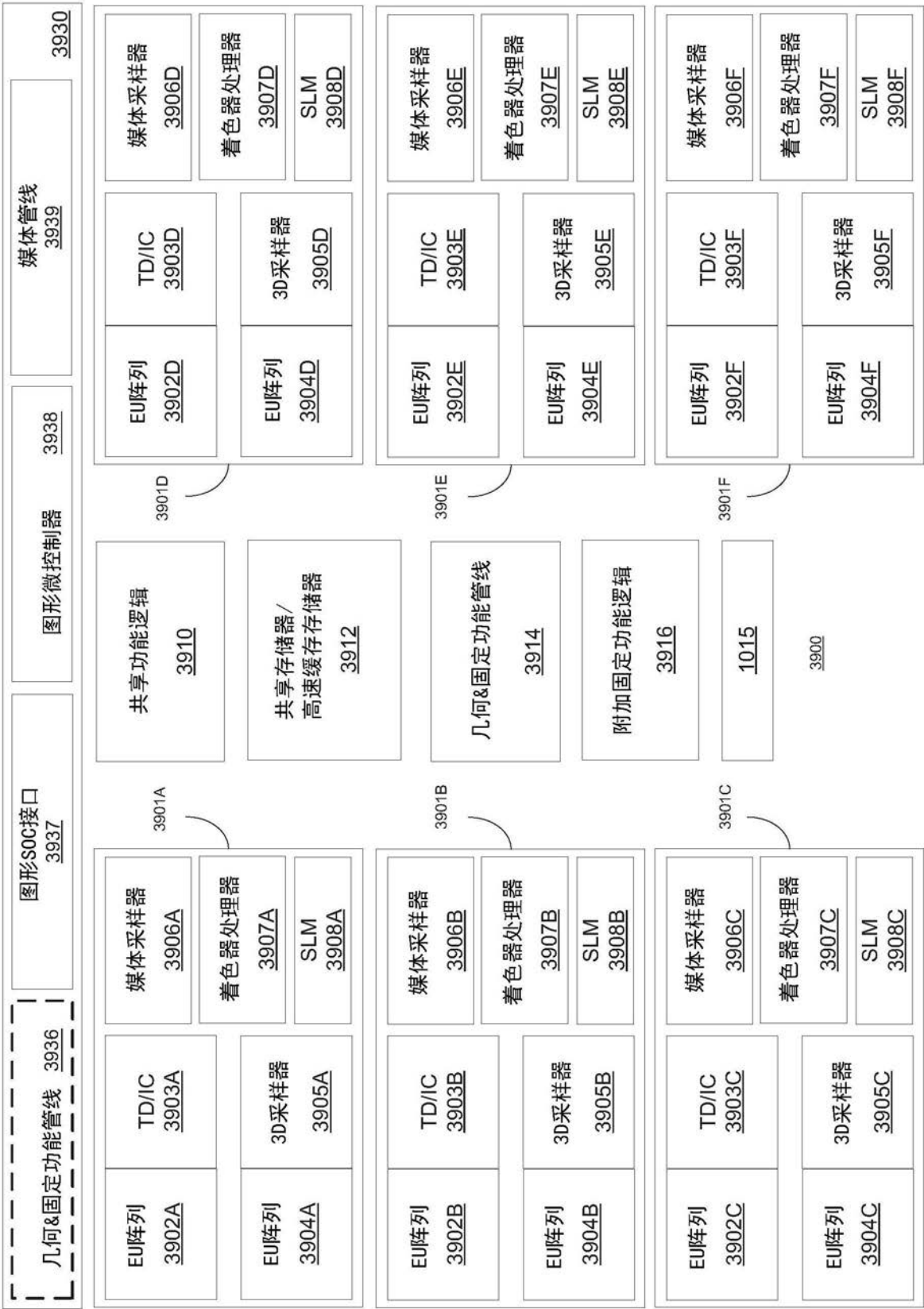


图39

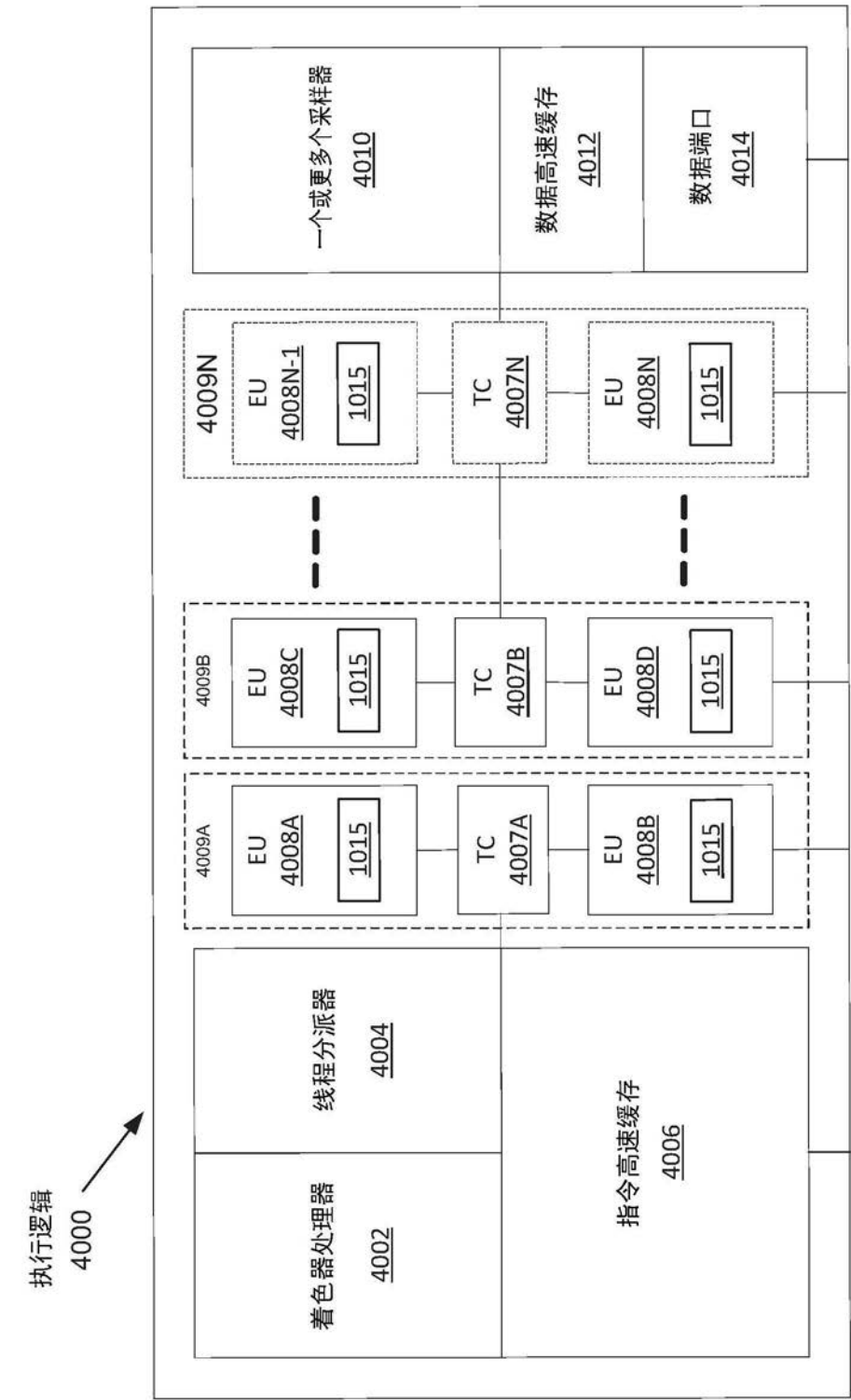


图40

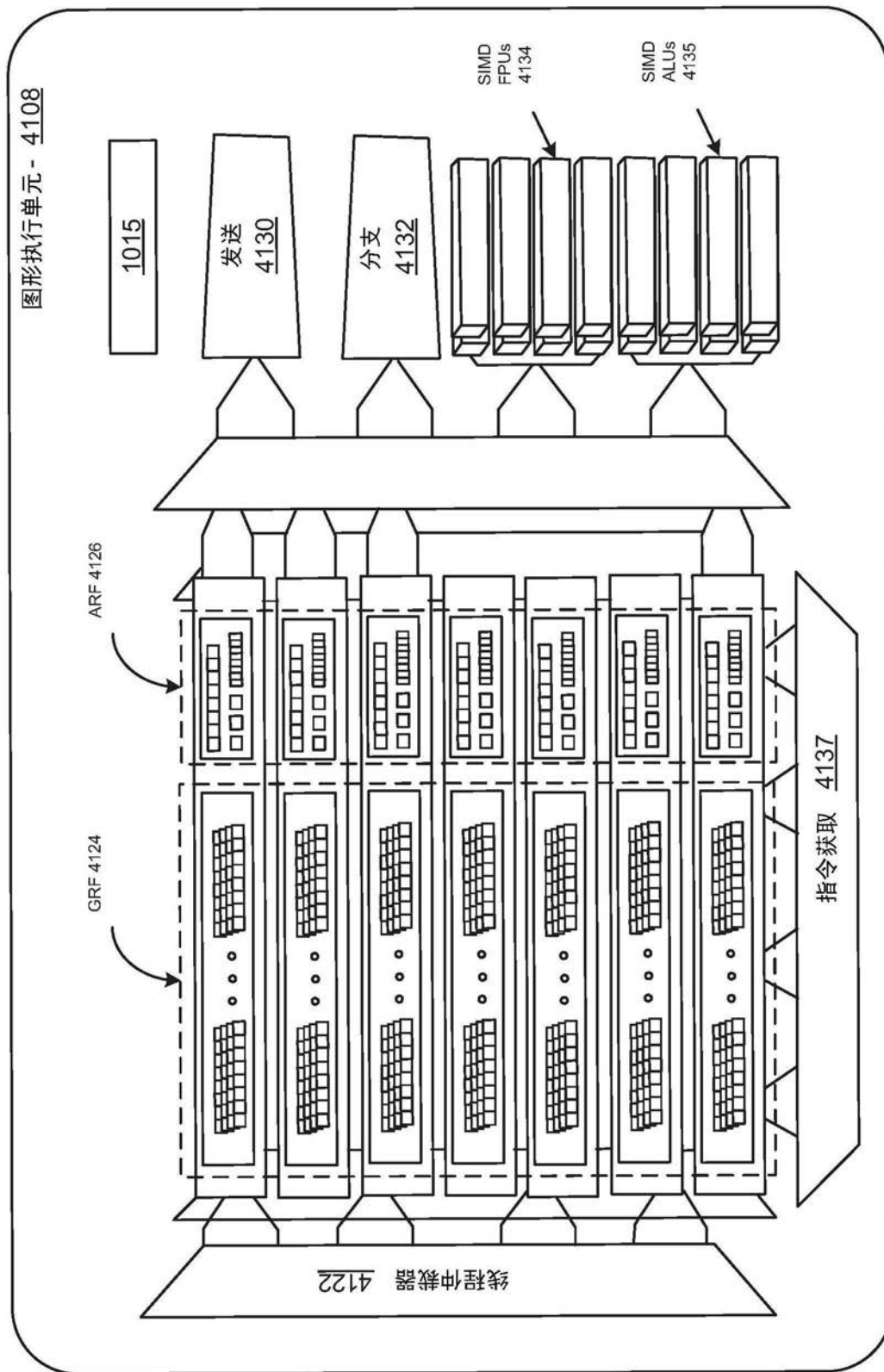


图41

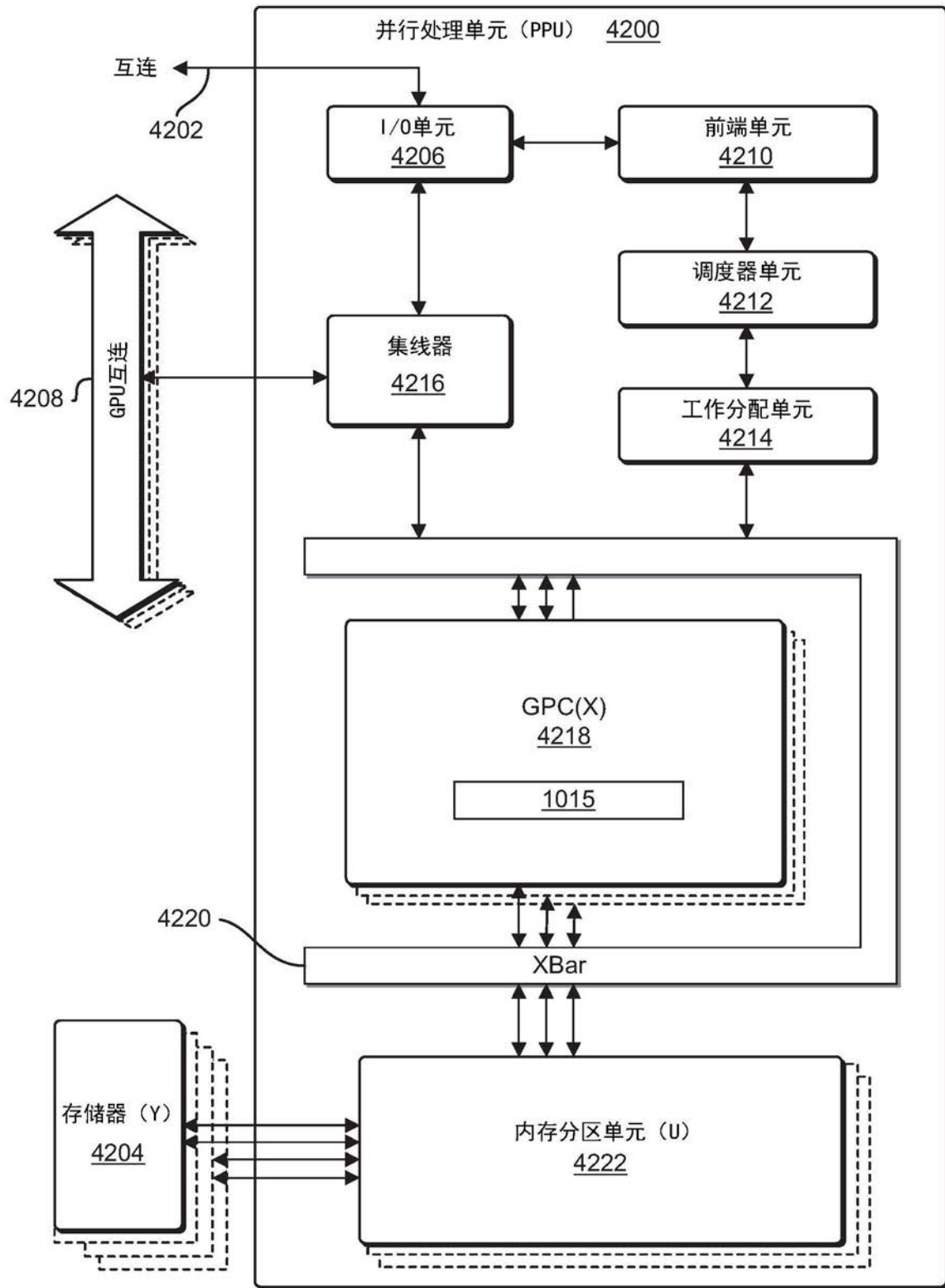


图42

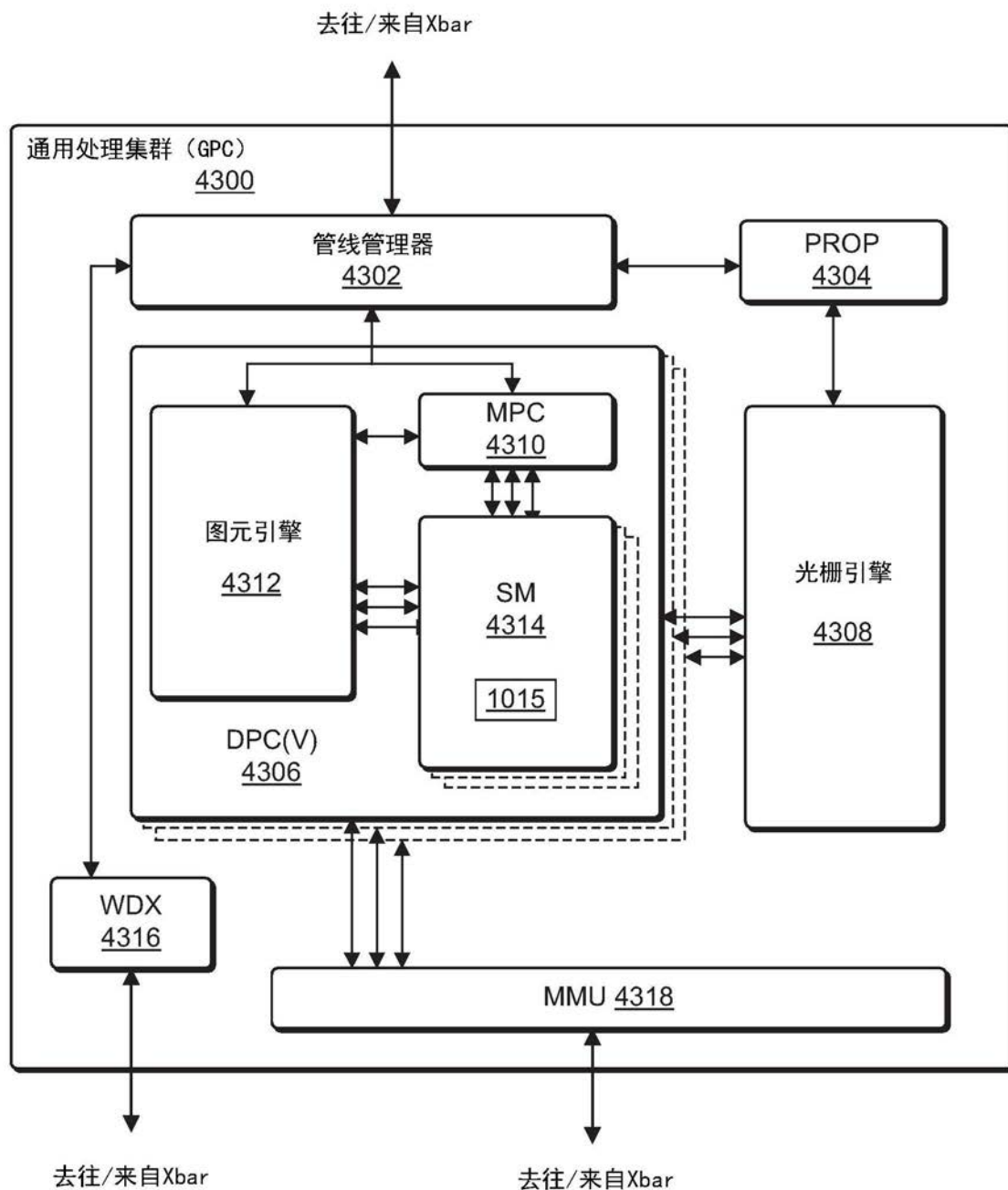


图43

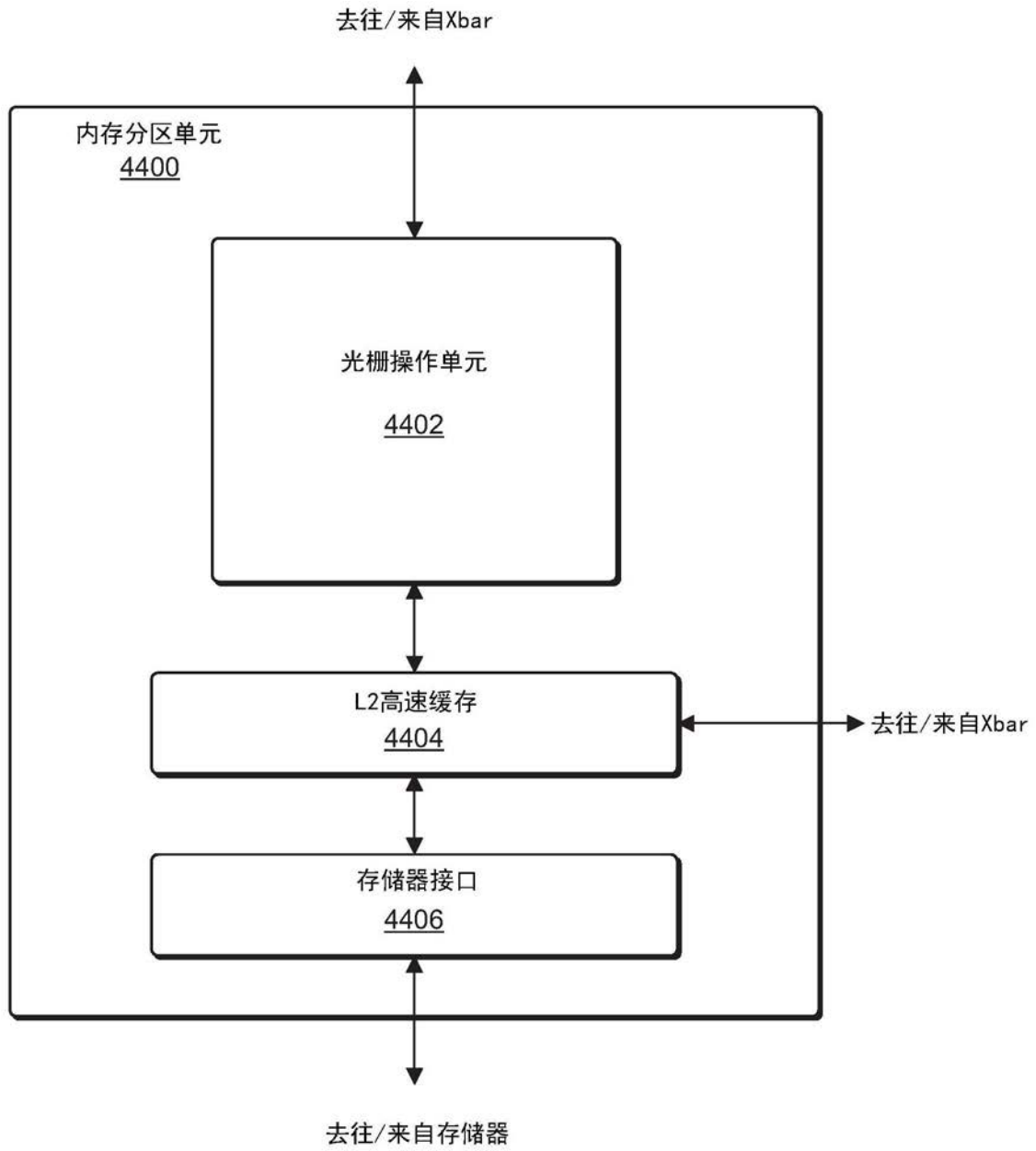


图44

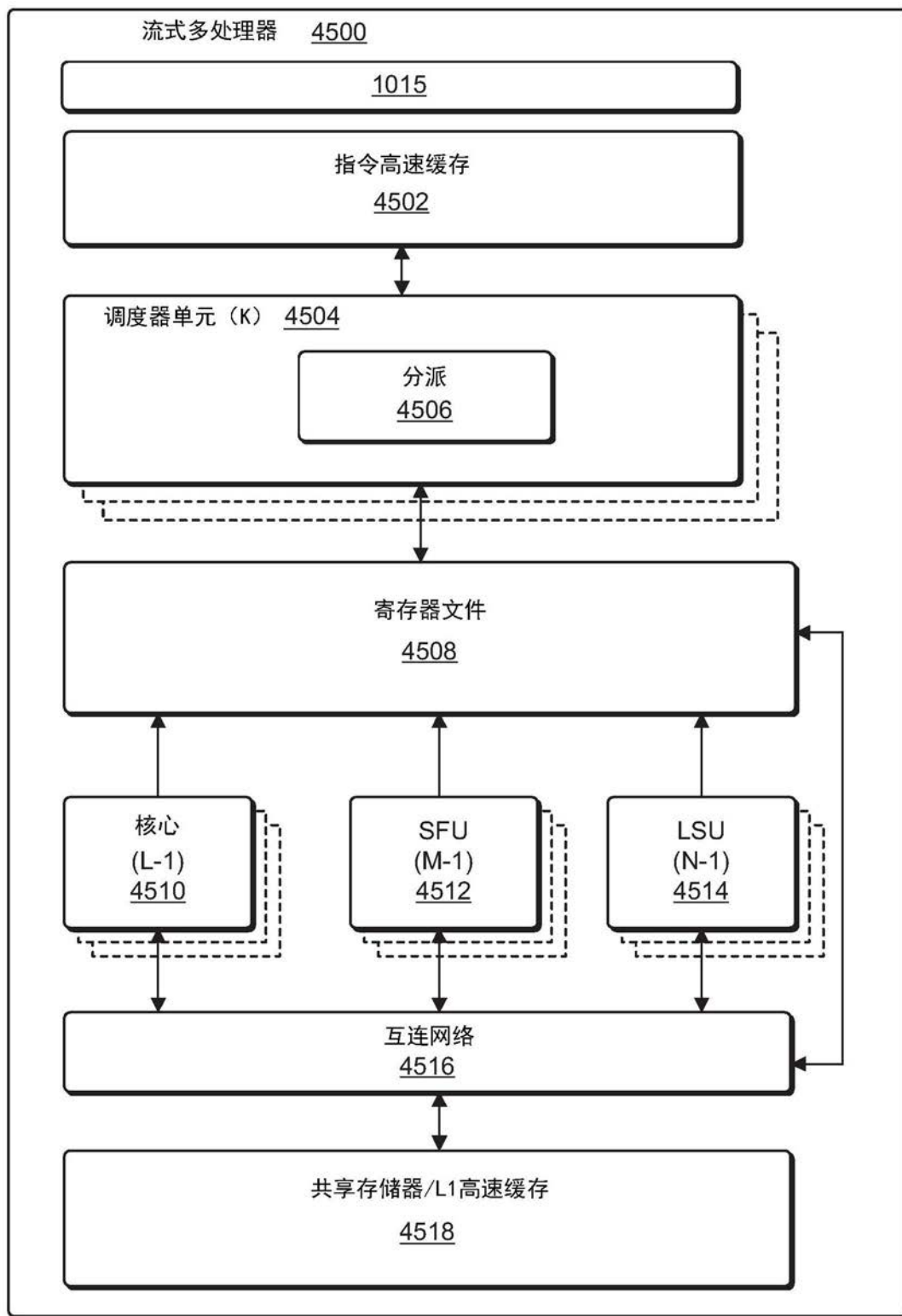


图45

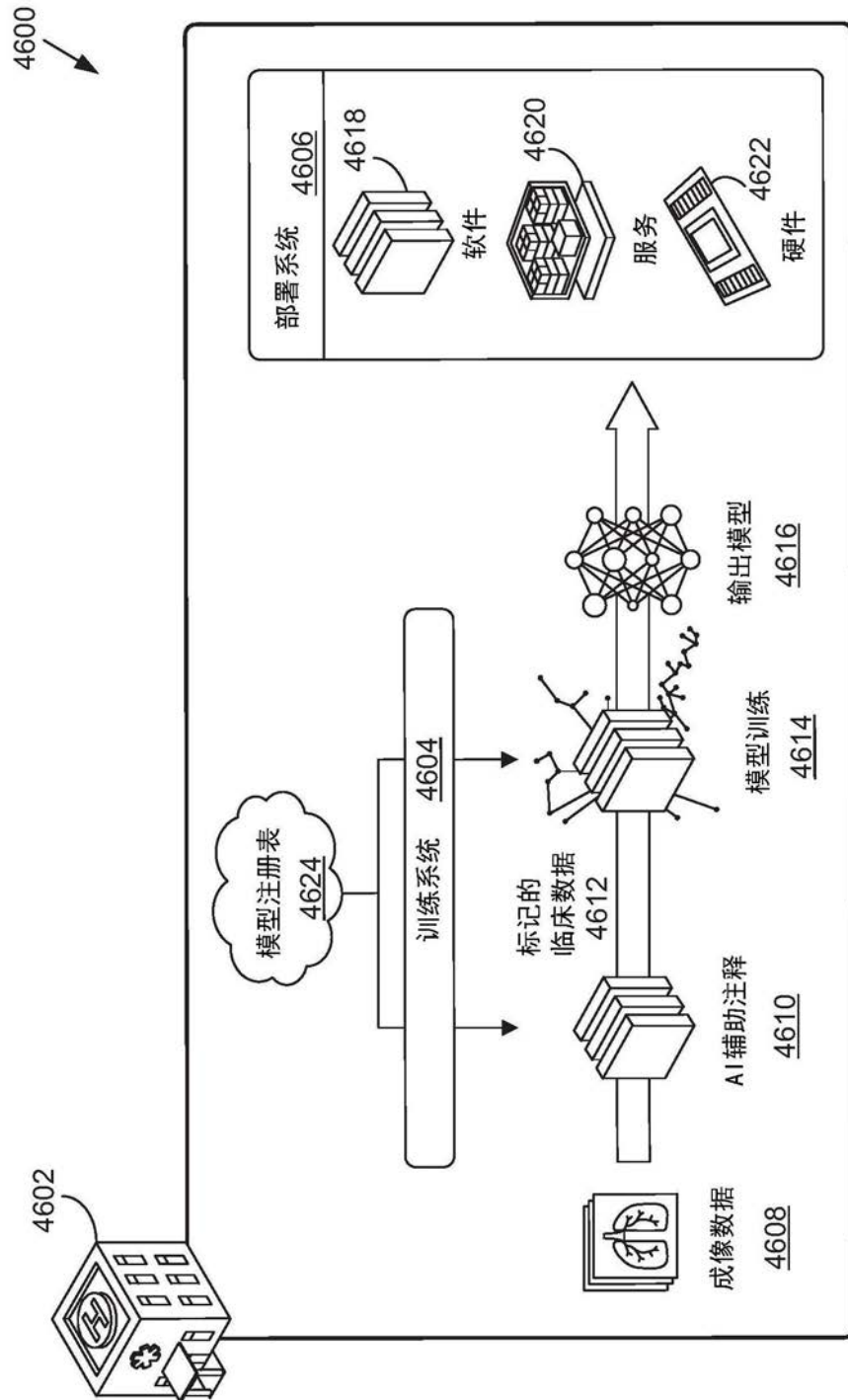


图46

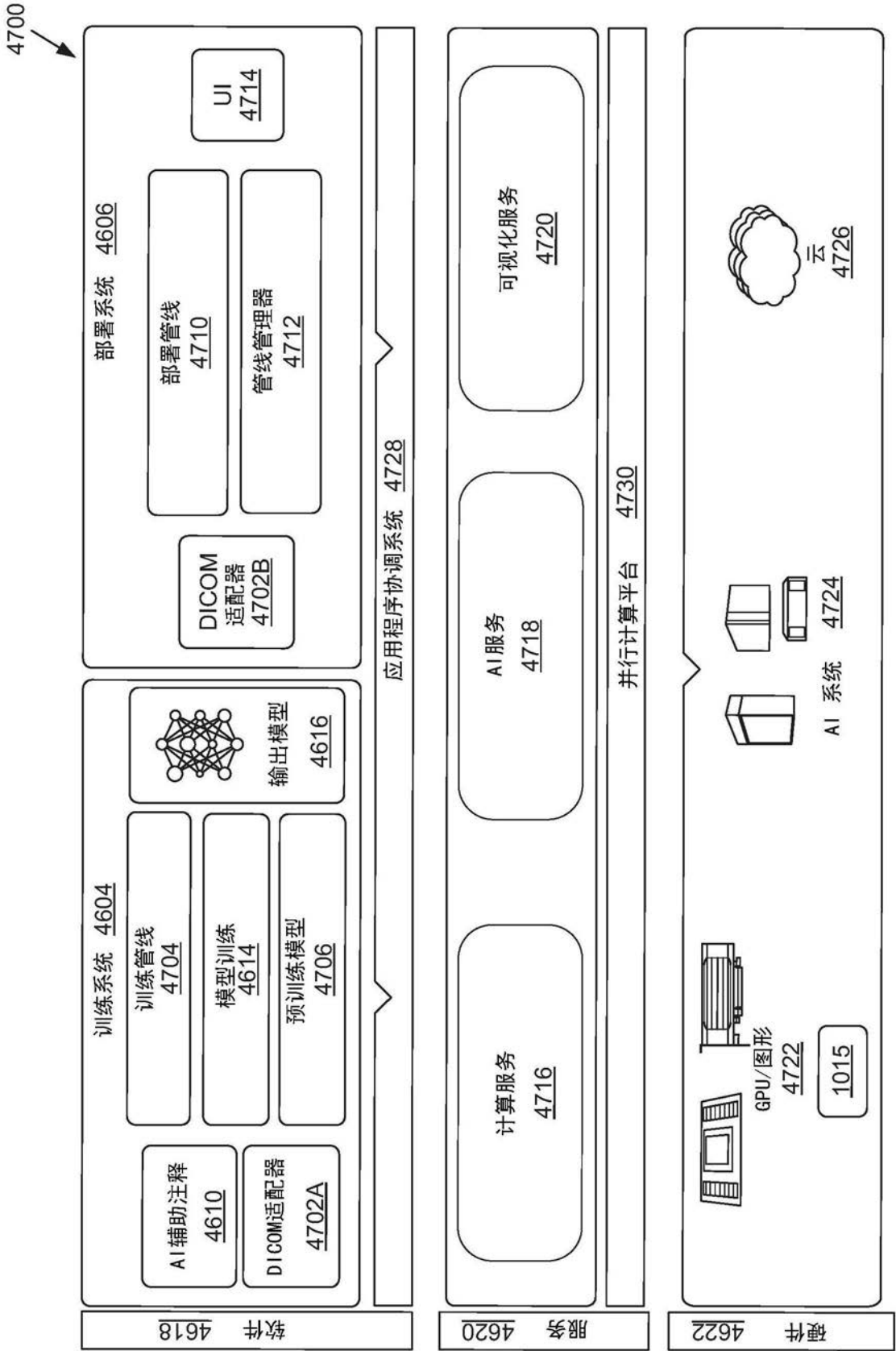


图47

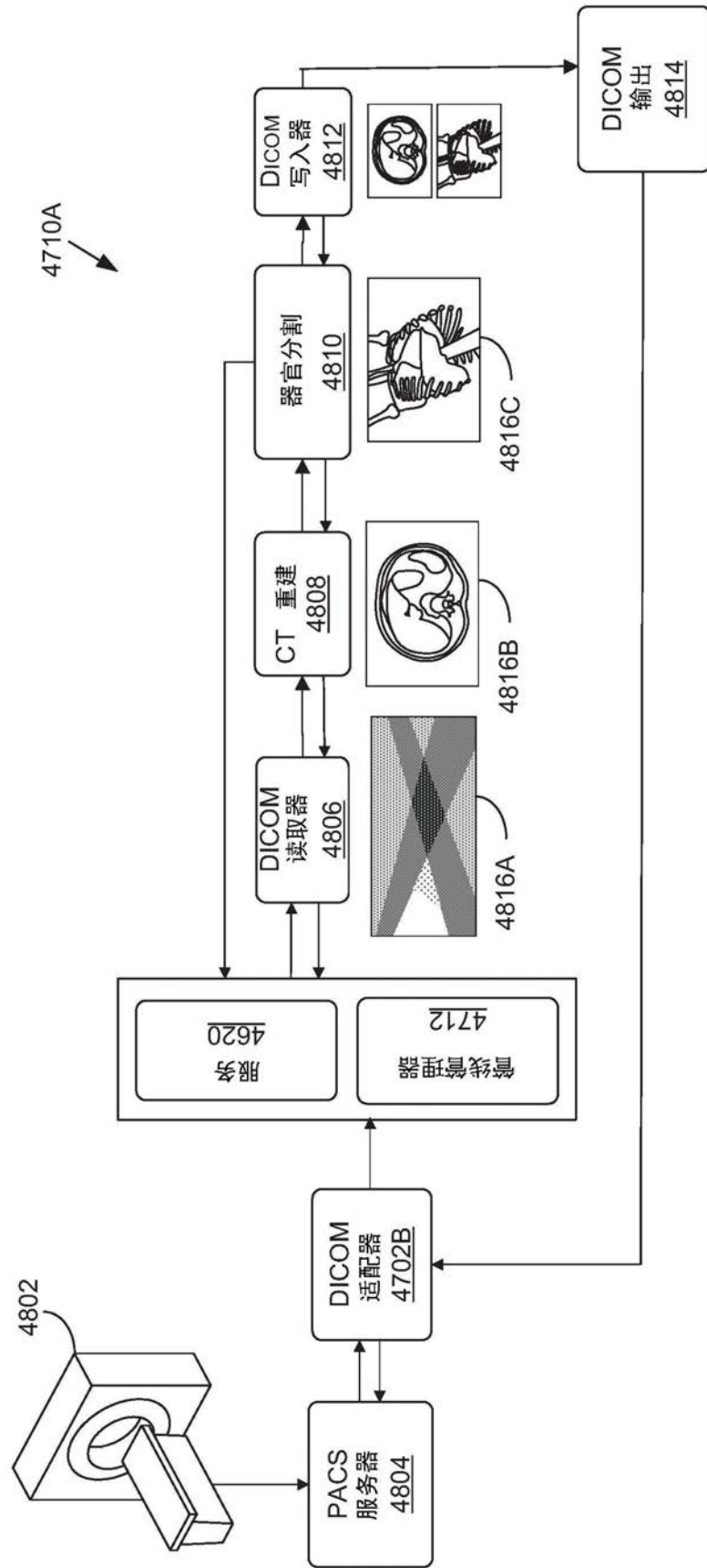


图48

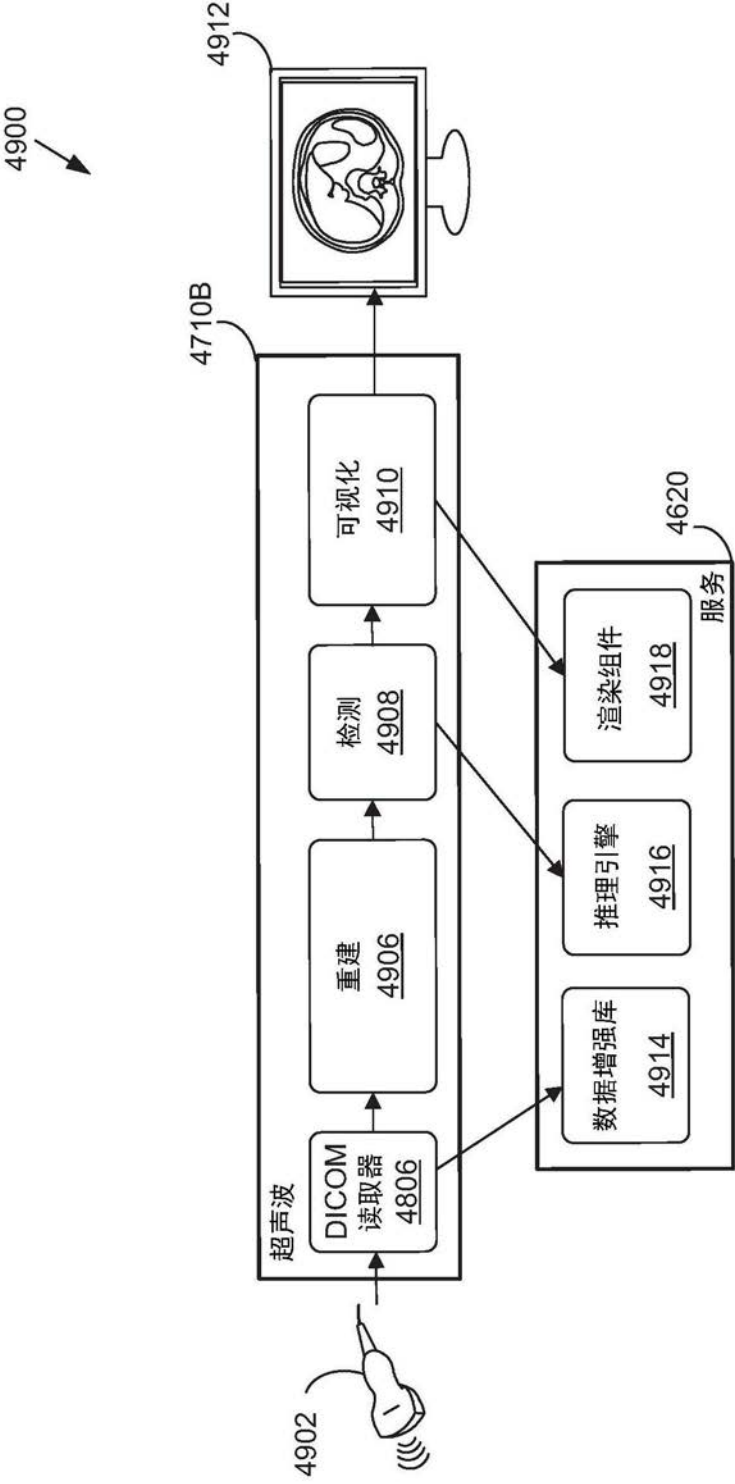


图49A

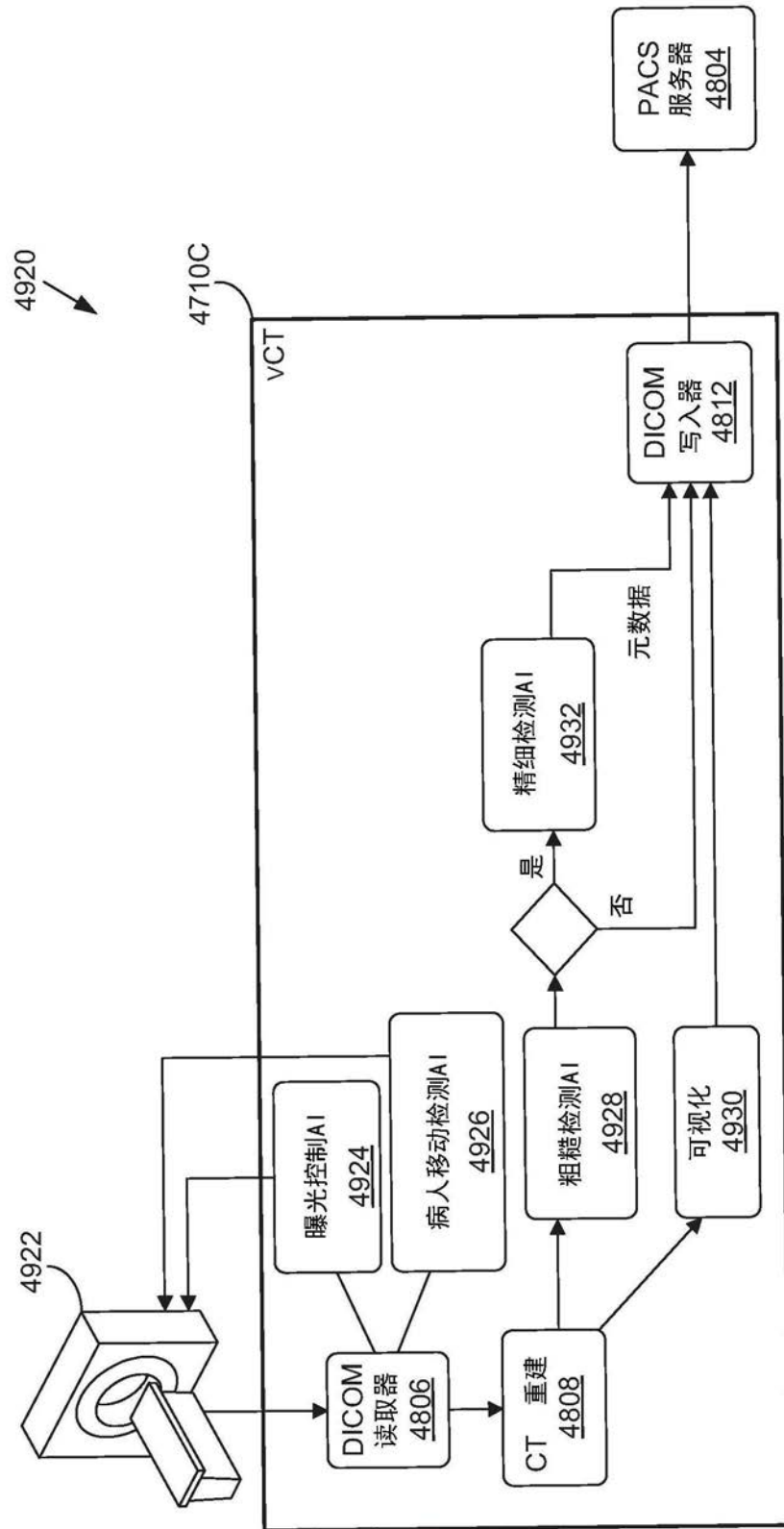


图49B

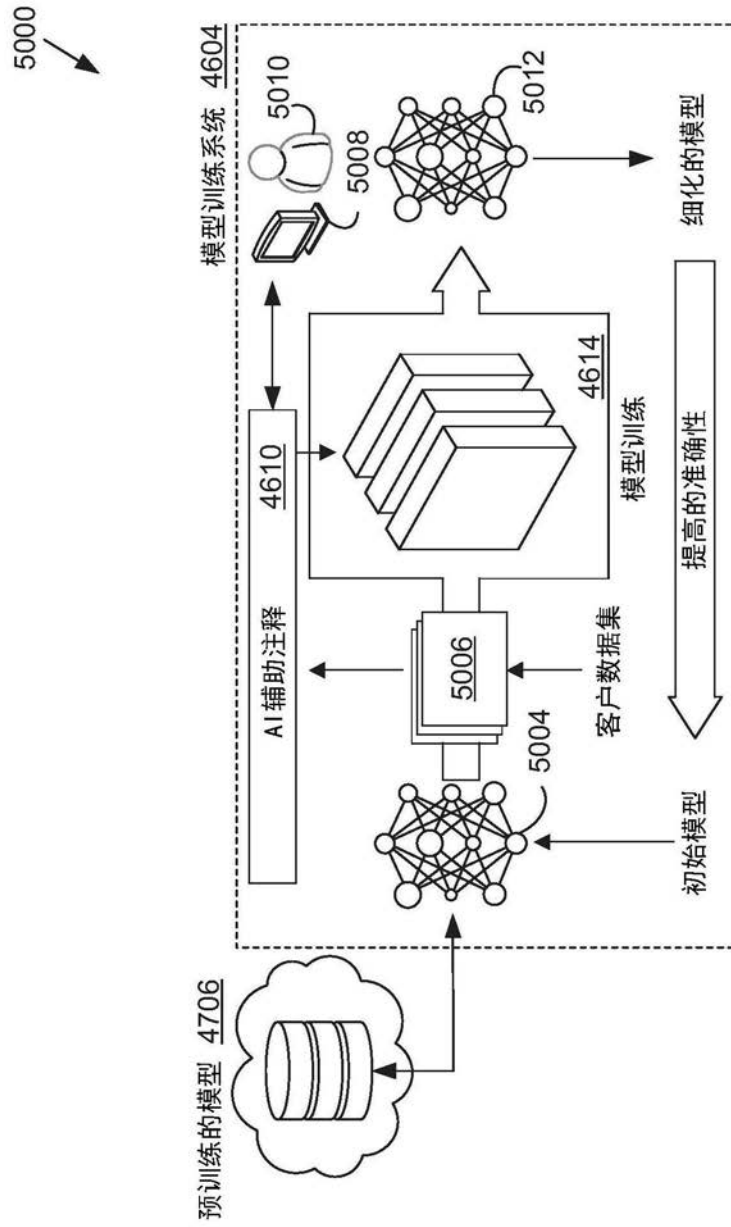


图50A

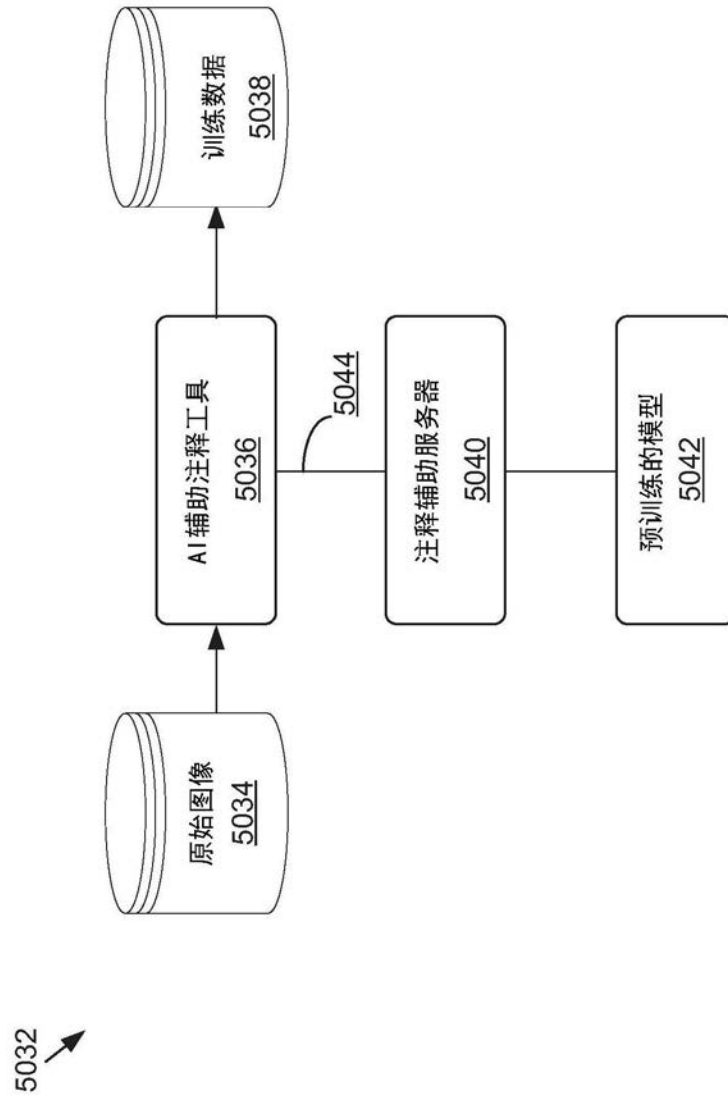


图50B