(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 113688089 A
(43)申请公布日 2021.11.23

(21)申请号 202111245073.3

(22)申请日 2021.10.26

(71)申请人 北京壁仞科技开发有限公司
地址 100085 北京市海淀区上地信息路26号1层0106-508室
申请人 上海壁仞智能科技有限公司

(72)发明人 不公告发明人

(74)专利代理机构 北京市金杜律师事务所 11256
代理人 王茂华

(51)Int.Cl.
G06F 15/16(2006.01)
G06N 3/04(2006.01)
G06N 3/063(2006.01)

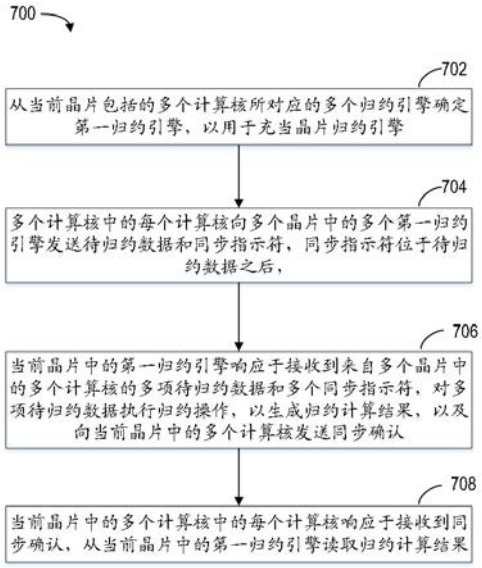
权利要求书2页 说明书8页 附图7页

(54)发明名称

数据处理方法、计算系统和计算机存储介质

(57)摘要

本公开的实施例涉及数据处理方法、计算系统和计算机存储介质。根据该方法，对于多个晶片中的每个晶片：从当前晶片包括的多个计算核所对应的多个归约引擎确定第一归约引擎，以用于充当晶片归约引擎；每个计算核向多个晶片中的多个第一归约引擎发送待归约数据和同步指示符，同步指示符位于待归约数据之后；当前晶片中的第一归约引擎响应于接收到来自多个晶片中的多个计算核的多项待归约数据和多个同步指示符，对多项待归约数据执行归约操作，以生成归约计算结果，以及向当前晶片中的多个计算核发送同步确认；以及当前晶片中的每个计算核响应于接收到同步确认，从当前晶片中的第一归约引擎读取归约计算结果。由此，能降低跨晶片的归约数据处理时延。



CN 113688089 A

1.一种数据处理方法,包括:

对于多个晶片中的每个晶片,执行以下步骤:

从当前晶片包括的多个计算核所对应的多个归约引擎确定第一归约引擎,以用于充当晶片归约引擎;

所述多个计算核中的每个计算核向所述多个晶片中的多个第一归约引擎发送待归约数据和同步指示符,所述同步指示符位于所述待归约数据之后;

当前晶片中的第一归约引擎响应于接收到来自所述多个晶片中的多个计算核的多项待归约数据和多个同步指示符,对所述多项待归约数据执行归约操作,以生成归约计算结果,以及向当前晶片中的所述多个计算核发送同步确认;以及

当前晶片中的所述多个计算核中的每个计算核响应于从当前晶片中的第一归约引擎接收到所述同步确认,从当前晶片中的第一归约引擎读取所述归约计算结果。

2.根据权利要求1所述的数据处理方法,其中所述多个晶片位于一个或多个计算装置中。

3.根据权利要求1所述的数据处理方法,其中所述多项待归约数据用于神经网络模型的批处理规范化。

4.根据权利要求3所述的数据处理方法,还包括:

对于多个晶片中的每个晶片,执行以下步骤:

当前晶片中的所述多个计算核中的每个计算核对所述归约计算结果执行批处理规范化。

5.根据权利要求1所述的数据处理方法,其中当前晶片中的第一归约引擎的数量为一个或多于一个。

6.一种计算系统,包括:

多个晶片,所述多个晶片中的每个晶片包括多个计算核,所述多个计算核中的每个计算核包括归约引擎,对于所述多个晶片中的每个晶片:

所述多个计算核所对应的多个归约引擎中的第一归约引擎被配置成充当晶片归约引擎;

所述多个计算核中的每个计算核被配置成向所述多个晶片中的多个第一归约引擎发送待归约数据和同步指示符,所述同步指示符位于所述待归约数据之后;

当前晶片中的第一归约引擎被配置成响应于接收到来自所述多个晶片中的多个计算核的多项待归约数据和多个同步指示符,对所述多项待归约数据执行归约操作,以生成归约计算结果,以及向当前晶片中的所述多个计算核发送同步确认;以及

当前晶片中的所述多个计算核中的每个计算核还被配置成响应于从当前晶片的第一归约引擎接收到所述同步确认,从当前晶片中的第一归约引擎读取所述归约计算结果。

7.根据权利要求6所述的计算系统,还包括一个或多个计算装置,所述多个晶片位于所述一个或多个计算装置中。

8.根据权利要求6所述的计算系统,其中所述多项待归约数据用于神经网络模型的批处理规范化。

9.根据权利要求8所述的计算系统,其中对于多个晶片中的每个晶片:

当前晶片中的所述多个计算核中的每个计算核还被配置成对所述归约计算结果执行

批处理规范化。

10.根据权利要求6所述的计算系统,其中当前晶片中的第一归约引擎的数量为一个或多于一个。

11.一种存储有计算机指令的非瞬时计算机可读存储介质,其特征在于,所述计算机指令用于使所述计算机执行权利要求1-5中任一项所述的方法。

数据处理方法、计算系统和计算机存储介质

技术领域

[0001] 本公开的实施例总体涉及信息处理领域，具体涉及数据处理方法、计算系统和计算机存储介质。

背景技术

[0002] 在多计算核、多晶片(die)乃至多计算装置(例如芯片)的计算系统中，深度学习任务批处理(batch)被划分并且在多计算核、多晶片乃至多计算装置间并行处理。具体来说，深度神经网络的每一次迭代，输入数据被等分成多份，然后分别在多个计算核、多个晶片乃至多个计算装置中独立进行前向传播和后向传播的运算，计算出梯度。在该次迭代完成后，合并梯度以及更新深度神经网络的参数，然后进行下一次迭代。

[0003] 深度神经网络包括多个隐含层。在训练过程中，各隐含层参数不停变化，导致每一层的输入分布会发生变化，使得每一层的输入分布不再满足独立同分布假设(IID，Independent and identically distributed)，也就是所谓的内部协变量漂移(ICS，Internal Covariate Shift)问题。为了解决内部协变量漂移问题，提出了批处理规范化(Batch Normalization，BN，也可以称为批处理标准化或批处理归一化)，即通过一定的规范化手段，使得每层神经网络的输入分布的均值为0且方差为1。

[0004] 在改进内部协变量漂移的同时，批处理规范化(batch normalization)也带了巨大的挑战，这是因为如果同步和规范化在远端晶片乃至在远端计算装置中的计算核中执行，由于来回地传输同步信息和数据而导致高同步开销与长时延，从而导致了严重的性能问题。

[0005] 在传统方案中，存在一个全局同步单元和全局归约(reduction)引擎。所有晶片或所有计算装置中的所有计算核均向全局同步单元发送同步指示符，以及发送待归约数据以在经编程的全局归约引擎中进行归约处理。通常，每个参与的计算核执行以下步骤，以进行批处理规范化：1)写出规范化数据以进行归约；2)将数据刷新出去；3)将同步指示符发送到全局同步单元；4)全局同步单元执行同步并且向计算核进行同步确认；5)读回归约计算的结果数据；6)进行数据规范化。

[0006] 可见，数据刷新、同步以及数据读取能够在跨晶片或跨装置的远端处发生，而这些操作会产生三倍跨晶片乃至跨装置的长时延。这将极大地影响整体性能。如图1所示，如果全局同步单元110位于计算装置1，并且全局归约操作在计算装置1中的晶片1中的计算核1中的归约引擎120中执行，则对于计算装置2中的晶片2中的计算核而言，时延是相当长的，这是因为它将物理上跨越多个晶片以及多个计算装置进行刷新、同步和数据读/写。

发明内容

[0007] 提供了一种数据处理方法、计算系统以及计算机存储介质，能够降低跨晶片的归约数据处理时延。

[0008] 根据本公开的第一方面，提供了一种数据处理方法。该数据处理方法包括：对于多

个晶片中的每个晶片，执行以下步骤：从当前晶片包括的多个计算核所对应的多个归约引擎确定第一归约引擎，以用于充当晶片归约引擎；多个计算核中的每个计算核向多个晶片中的多个第一归约引擎发送待归约数据和同步指示符，同步指示符位于所述待归约数据之后；当前晶片中的第一归约引擎响应于接收到来自多个晶片中的多个计算核的多项待归约数据和多个同步指示符，对多项待归约数据执行归约操作，以生成归约计算结果，以及向当前晶片中的多个计算核发送同步确认；以及当前晶片中的多个计算核中的每个计算核响应于从当前晶片中的第一归约引擎接收到同步确认，从当前晶片中的第一归约引擎读取归约计算结果。

[0009] 根据本公开的第二方面，提供了一种计算系统。该计算系统包括：多个晶片，多个晶片中的每个晶片包括多个计算核，多个计算核中的每个计算核包括归约引擎，对于多个晶片中的每个晶片：多个计算核所对应的多个归约引擎中的第一归约引擎被配置成充当晶片归约引擎；多个计算核中的每个计算核被配置成向多个晶片中的多个第一归约引擎发送待归约数据和同步指示符，同步指示符位于待归约数据之后；当前晶片中的第一归约引擎被配置成响应于接收到来自多个晶片中的多个计算核的多项待归约数据和多个同步指示符，对多项待归约数据执行归约操作，以生成归约计算结果，以及向当前晶片中的多个计算核发送同步确认；以及当前晶片中的多个计算核中的每个计算核还被配置成响应于从当前晶片中的第一归约引擎接收到同步确认，从当前晶片中的第一归约引擎读取归约计算结果。

[0010] 在本公开的第三方面中，提供了一种计算机可读存储介质，其上存储有计算机程序，该程序被处理器执行时实现根据本公开的第一方面的方法。

[0011] 应当理解，本部分所描述的内容并非旨在标识本公开的实施例的关键或重要特征，也不用于限制本公开的范围。本公开的其它特征将通过以下的说明书而变得容易理解。

附图说明

[0012] 结合附图并参考以下详细说明，本公开各实施例的上述和其他特征、优点及方面将变得更加明显。在附图中，相同或相似的附图标注表示相同或相似的元素。

[0013] 图1是根据现有技术的计算系统100的示意图。

[0014] 图2是根据本公开的实施例的计算系统200的示意图。

[0015] 图3是根据本公开的实施例的数据传输过程300的示意图。

[0016] 图4是根据本公开的实施例的数据传输过程400的示意图。

[0017] 图5是根据本公开的实施例的数据传输过程500的示意图。

[0018] 图6是根据本公开的实施例的数据传输过程600的示意图。

[0019] 图7是根据本公开的实施例的数据处理方法700的示意图。

具体实施方式

[0020] 以下结合附图对本公开的示范性实施例做出说明，其中包括本公开实施例的各种细节以助于理解，应当将它们认为仅仅是示范性的。因此，本领域普通技术人员应当认识到，可以对这里描述的实施例做出各种改变和修改，而不会背离本公开的范围和精神。同样，为了清楚和简明，以下的描述中省略了对公知功能和结构的描述。

[0021] 在本文中使用的术语“包括”及其变形表示开放性包括，即“包括但不限于”。除非特别申明，术语“或”表示“和/或”。术语“基于”表示“至少部分地基于”。术语“一个示例实施例”和“一个实施例”表示“至少一个示例实施例”。术语“另一实施例”表示“至少一个另外的实施例”。术语“第一”、“第二”等等可以指代不同的或相同的对象。下文还可能包括其他明确的和隐含的定义。

[0022] 如上所述，当数据刷新、同步以及数据读取在跨晶片或跨装置的远端处发生时传统方案会产生三倍跨晶片乃至跨装置的长时延。这将极大地影响整体性能。

[0023] 为了至少部分地解决上述问题以及其他潜在问题中的一个或者多个，本公开的示例实施例提出了一种用于数据处理的方案。在该方案中，对于多个晶片中的每个晶片，执行以下步骤：从当前晶片包括的多个计算核所对应的多个归约引擎确定第一归约引擎，以用于充当晶片归约引擎；多个计算核中的每个计算核向多个晶片中的多个第一归约引擎发送待归约数据和同步指示符，同步指示符位于所述待归约数据之后；当前晶片中的第一归约引擎响应于接收到来自多个晶片中的多个计算核的多项待归约数据和多个同步指示符，对多项待归约数据执行归约操作，以生成归约计算结果，以及向当前晶片中的多个计算核发送同步确认；以及当前晶片中的多个计算核中的每个计算核响应于从当前晶片中的第一归约引擎接收到同步确认，从当前晶片中的第一归约引擎读取归约计算结果。

[0024] 以此方式，能够降低跨晶片的归约数据处理时延。

[0025] 在下文中，将结合附图更详细地描述本方案的具体示例。

[0026] 图2示出了根据本公开的实施例的计算系统200的示例的示意图。如图2所示，计算系统200可以包括4个计算装置210、220、230和240。应当理解，虽然图2中示出了4个计算装置，但是这只是举例说明，计算设备200可以包括更多或更少的计算装置，例如1、2、3、5个计算装置等。

[0027] 计算装置210-240可以包括多个晶片。如图2所示，计算装置210包括2个晶片211和212，计算装置220包括2个晶片221和222，计算装置230包括2个晶片231和232，以及计算装置240包括2个晶片241和242。应当理解，虽然图2中示出了计算装置包括2个晶片，但是这只是举例说明，计算装置可以包括更多的晶片。

[0028] 计算装置210-240例如包括但不限于封装有多个晶片的芯片(chip)，例如中央处理器芯片(CPU芯片)、图形处理器芯片(GPU芯片)、人工智能芯片(AI芯片)、高性能计算芯片(HPC芯片)等。计算装置内的多个晶片之间可以通过高速IO总线连接各个晶片内的片上网络(NOC,Network On Chip)进行连接。计算装置之间可以通过芯片间互联技术进行连接，例如Blink或者高速IO总线。

[0029] 晶片可以包括多个计算核。计算核可以被配置成执行一个或多个通用或定制操作。例如，计算核可以被配置成执行逻辑、对数、指数、乘法、比较、三角函数、矩阵运算和/或其他适当的通用操作。作为备选或者附加，计算核可以被配置成执行神经网络模型训练和推理、系统发育推断、基因组测序、气候建模、天气预测、视频/声音处理和/或其它适当的定制操作。计算核可以包括归约引擎(也可以称为归约计算单元)。归约引擎可以被配置成执行归约操作，归约操作例如包括但不限于加和、乘积等。

[0030] 如图2所示，晶片211、212、221、222、231、232、241以及242各包括n个计算核，其中n大于或等于2。应当理解，虽然图2中示出了晶片具有相同数量的计算核，但是这只是举例说

明，不同晶片也可以具有不同数量的计算核。

[0031] 对于每个晶片而言，多个计算核所对应的多个归约引擎中的第一归约引擎(例如，计算核1中的归约引擎)可以被配置为充当晶片归约引擎。多个计算核中的每个计算核(例如，计算核1至计算核n)可以被配置成向多个晶片(例如晶片211、212、221、222、231、232、241以及242)中的多个第一归约引擎(例如，晶片211、212、221、222、231、232、241以及242中的计算核1中的归约引擎)发送待归约数据和同步指示符，同步指示符位于待归约数据之后。这里的多个晶片可以位于一个或多个计算装置中。例如，在计算系统200包括一个计算装置的情况下，这里的多个晶片可以位于一个计算装置中。例如，在计算系统200包括多个计算装置的情况下，这里的多个晶片可以位于多个计算装置中。待归约数据例如可用于神经网络模型的批处理规范化。

[0032] 图3示出了根据本公开的实施例的数据传输过程的示意图。如图3所示，晶片311和312位于计算装置310中，晶片321和322位于计算装置320中，晶片331和332位于计算装置330中，以及晶片341和342位于计算装置340中。晶片311、312、321、322、331、332、341以及342中的计算核1中的归约引擎均可以被配置为充当晶片归约引擎。应当理解，这只是举例说明，晶片中的其他计算核中的归约引擎也可以被配置为充当晶片归约引擎，本公开的范围在此不受限制。还应当理解，虽然图3中仅示出了晶片中的一个归约引擎被配置为充当晶片归约引擎，但是这只是举例说明，晶片中多于一个归约引擎(例如，计算核1中的归约引擎和计算核n中的归约引擎)可以被配置为充当晶片归约引擎，也就是说，一个晶片中被配置为充当晶片归约引擎的第一归约引擎的数量可以是一个或多于一个，本公开的范围在此不受限制。

[0033] 晶片311中的计算核1至计算核n可以被配置成向晶片311、312、321、322、331、332、341以及342中的8个计算核1中的8个归约引擎发送待归约数据和同步指示符。也就是说，晶片311中的计算核1至计算核n既向相同晶片311中的计算核1中充当晶片归约引擎的归约引擎发送待归约数据和同步指示符，也向其他晶片312、321、322、331、332、341以及342中的计算核1中充当晶片归约引擎的归约引擎发送待归约数据和同步指示符。这里应当注意，晶片311中充当晶片归约引擎的归约引擎所在的计算核1也向该计算核1中的归约引擎发送了待归约数据和同步指示符。

[0034] 应当理解，为了清楚展示，图3中仅示出了晶片311中的计算核1和计算核n向311、312、321、322、331、332、341以及342中的8个计算核1中的8个归约引擎发送待归约数据和同步指示符，但是其他计算核对于待归约数据和同步指示符的传输也是类似的。

[0035] 类似地，晶片312、321、322、331、332、341以及342中的计算核1至计算核n也可以被配置成向晶片311、312、321、322、331、332、341以及342中的8个计算核1中的8个归约引擎发送待归约数据和同步指示符(未示出)。

[0036] 对于每个晶片而言，当前晶片中的第一归约引擎可以被配置成响应于接收到来自多个晶片中的多个计算核的多项待归约数据和多个同步指示符，对多项待归约数据执行归约操作，以生成归约计算结果，以及向当前晶片中的多个计算核发送同步确认。

[0037] 图4示出了根据本公开的实施例的数据传输过程的示意图。

[0038] 如图4所示，晶片411和412位于计算装置410中，晶片421和422位于计算装置420中，晶片431和432位于计算装置430中，以及晶片441和442位于计算装置440中。晶片411、

412、421、422、431、432、441以及442中的计算核1中的归约引擎均可以被配置为充当晶片归约引擎。应当理解，这只是举例说明，晶片中的其他计算核中的归约引擎也可以被配置为充当晶片归约引擎，本公开的范围在此不受限制。还应当理解，虽然图4中仅示出了晶片中的一个归约引擎被配置为充当晶片归约引擎，但是这只是举例说明，晶片中多于一个归约引擎(例如，计算核1中的归约引擎和计算核n中的归约引擎)可以被配置为充当晶片归约引擎，也就是说，一个晶片中被配置为充当晶片归约引擎的第一归约引擎的数量可以是一个或多于一个，本公开的范围在此不受限制。

[0039] 晶片411中的计算核1中充当晶片归约引擎的归约引擎可以接收来自晶片411、412、421、422、431、432、441以及442中的计算核1至计算核n的多项待归约数据(一共8*n项)和多项同步指示符。晶片411、412、421、422、431、432、441以及442中的计算核1中的归约引擎也是类似，不再赘述。

[0040] 图5示出了根据本公开的实施例的数据传输过程的示意图。

[0041] 如图5所示，晶片511和512位于计算装置510中，晶片521和522位于计算装置520中，晶片531和532位于计算装置530中，以及晶片541和542位于计算装置540中。晶片511、512、521、522、531、532、541以及542中的计算核1中的归约引擎均可以被配置为充当晶片归约引擎。应当理解，这只是举例说明，晶片中的其他计算核中的归约引擎也可以被配置为充当晶片归约引擎，本公开的范围在此不受限制。还应当理解，虽然图5中仅示出了晶片中的一个归约引擎被配置为充当晶片归约引擎，但是这只是举例说明，晶片中多于一个归约引擎(例如，计算核1中的归约引擎和计算核n中的归约引擎)可以被配置为充当晶片归约引擎，也就是说，一个晶片中被配置为充当晶片归约引擎的第一归约引擎的数量可以是一个或多于一个，本公开的范围在此不受限制。

[0042] 晶片511中的计算核1中充当晶片归约引擎的归约引擎可以对接收的多项待归约数据(一共8*n项)执行归约操作，以生成归约计算结果，以及向晶片511中的计算核1至计算核n发送同步确认。也就是说，充当晶片归约引擎的归约引擎接收来自本地晶片和远端晶片中的多个计算核的多项待归约数据和多项同步指示符，但是仅向本地晶片中的多个计算核发送同步确认，这就是分布式归约同步和计算的具体表现。晶片511、512、521、522、531、532、541以及542中的计算核1中的归约引擎也是类似，不再赘述。

[0043] 对于每个晶片而言，当前晶片中的多个计算核中的每个计算核可以被配置成响应于从当前晶片中的第一归约引擎接收到同步确认，从当前晶片中的第一归约引擎读取归约计算结果。

[0044] 图6示出了根据本公开的实施例的数据传输过程的示意图。如图6所示，晶片611和612位于计算装置610中，晶片621和622位于计算装置620中，晶片631和632位于计算装置630中，以及晶片641和642位于计算装置640中。晶片611、612、621、622、631、632、641以及642中的计算核1中的归约引擎均可以被配置为充当晶片归约引擎。应当理解，这只是举例说明，晶片中的其他计算核中的归约引擎也可以被配置为充当晶片归约引擎，本公开的范围在此不受限制。还应当理解，虽然图6中仅示出了晶片中的一个归约引擎被配置为充当晶片归约引擎，但是这只是举例说明，晶片中多于一个归约引擎(例如，计算核1中的归约引擎和计算核n中的归约引擎)可以被配置为充当晶片归约引擎，也就是说，一个晶片中被配置为充当晶片归约引擎的第一归约引擎的数量可以是一个或多于一个，本公开的范围在此不

受限制。

[0045] 晶片611中的计算核1至计算核n可以响应于接收到来自晶片611中的计算核1中的归约引擎的同步确认，从晶片611中的计算核1中的归约引擎读取归约计算结果。也就是说，晶片中的多个计算核仅从本地晶片中充当晶片归约引擎的归约引擎接收同步确认并读取归约计算结果，而无需从远端晶片接收同步确认并读取归约计算结果，从而节省了从远端晶片远距离获取上述信息的时延。晶片612、621、622、631、632、641以及642中的计算核1至计算核n也是类似，不再赘述。

[0046] 此外，对于每个晶片而言，当前晶片中的多个计算核中的每个计算核还可以被配置成对归约计算结果执行批处理规范化。

[0047] 由此，对于多个晶片中的每个晶片，该晶片中的多个计算核所对应的多个归约引擎中的第一归约引擎被确定用于充当晶片归约引擎，该晶片中的多个计算核向多个晶片中的多个第一归约引擎发送待归约数据和同步指示符，但是从本地晶片中的第一归约引擎接收同步确认并读取归约计算结果，而无需从远端晶片接收同步确认并读取归约计算结果，从而实现分布式的归约，极大地降低了跨晶片归约处理时延。虽然每个计算核发送多份待归约数据到不同的晶片乃至不同计算设备，这看起来增加了数据传输。但是由于归约一般是分级的，大部分归约已经在每个计算核内部进行了归约处理，因此更高级别的待归约数据传输已经得到了极大的减少，总体上不是大的问题。另外，同步指示符跟在对应的待归约数据之后，确保了在同步完成的时候，归约操作已经比同步更早完成，从而确保的数据一致性，无需全局的刷新指令。本公开方案能够实现3-4倍跨晶片或跨计算装置的归约处理时延降低，从而极大提升多计算核、多晶片乃至多计算装置的深度学习计算系统中批处理规范化的性能。

[0048] 图7示出了根据本公开的实施例的数据处理方法700的流程图。例如，方法700可以由如图1所示计算系统200针对其中多个晶片中的每个晶片而执行。应当理解的是，方法700还可以包括未示出的附加框和/或可以省略所示出的框，本公开的范围在此方面不受限制。

[0049] 在框702处，从当前晶片包括的多个计算核所对应的多个归约引擎确定第一归约引擎，以用于充当晶片归约引擎。

[0050] 第一归约引擎例如可以通过随机确定或者按照预定规则确定。当前晶片中被确定用于充当晶片归约引擎的第一归约引擎的数量可以为一个或多于一个。

[0051] 在框704处，多个计算核中的每个计算核向多个晶片中的多个第一归约引擎发送待归约数据和同步指示符，同步指示符位于待归约数据之后。

[0052] 待归约数据例如用于神经网络模型的批处理规范化。

[0053] 在框706处，当前晶片中的第一归约引擎响应于接收到来自多个晶片中的多个计算核的多项待归约数据和多个同步指示符，对多项待归约数据执行归约操作，以生成归约计算结果，以及向当前晶片中的多个计算核发送同步确认。

[0054] 归约操作例如包括但不限于加和、乘积等。

[0055] 在框708处，当前晶片中的多个计算核中的每个计算核响应于从当前晶片中的第一归约引擎接收到同步确认，从当前晶片中的第一归约引擎读取归约计算结果。

[0056] 此外，当前晶片中的多个计算核中的每个计算核还可以对归约计算结果执行批处理规范化。

[0057] 由此,对于多个晶片中的每个晶片,该晶片中的多个计算核所对应的多个归约引擎中的第一归约引擎被确定用于充当晶片归约引擎,该晶片中的多个计算核向多个晶片中的多个第一归约引擎发送待归约数据和同步指示符,但是从本地晶片中的第一归约引擎接收同步确认并读取归约计算结果,而无需从远端晶片接收同步确认并读取归约计算结果,从而实现分布式的归约,极大地降低了跨晶片归约处理时延。虽然每个计算核发送多份待归约数据到不同的晶片乃至不同计算设备,这看起来增加了数据传输。但是由于归约一般是分级的,大部分归约已经在每个计算核内部进行了归约处理,因此更高级别的待归约数据传输已经得到了极大的减少,总体上不是大的问题。另外,同步指示符跟在对应的待归约数据之后,确保了在同步完成的时候,归约操作已经比同步更早完成,从而确保的数据一致性,无需全局的刷新指令。本公开方案能够实现3-4倍跨晶片或跨计算装置的归约处理时延降低,从而极大提升多计算核、多晶片乃至多计算装置的深度学习计算系统中批处理规范化的性能。

[0058] 本公开涉及方法、装置、系统、计算设备、计算机可读存储介质和/或计算机程序产品。计算机程序产品可以包括用于执行本公开的各个方面的计算机可读程序指令。

[0059] 计算机可读存储介质可以是可以保持和存储由指令执行设备使用的指令的有形设备。计算机可读存储介质例如可以是——但不限于——电存储设备、磁存储设备、光存储设备、电磁存储设备、半导体存储设备或者上述的任意合适的组合。计算机可读存储介质的更具体的例子(非穷举的列表)包括:便携式计算机盘、硬盘、随机存取存储器(RAM)、只读存储器(ROM)、可擦式可编程只读存储器(EPROM或闪存)、静态随机存取存储器(SRAM)、便携式压缩盘只读存储器(CD-ROM)、数字多功能盘(DVD)、记忆棒、软盘、机械编码设备、例如其上存储有指令的打孔卡或凹槽内凸起结构、以及上述的任意合适的组合。这里所使用的计算机可读存储介质不被解释为瞬时信号本身,诸如无线电波或者其他自由传播的电磁波、通过波导或其他传输媒介传播的电磁波(例如,通过光纤电缆的光脉冲)、或者通过电线传输的电信号。

[0060] 这里所描述的计算机可读程序指令可以从计算机可读存储介质下载到各个计算/处理设备,或者通过网络、例如因特网、局域网、广域网和/或无线网下载到外部计算机或外部存储设备。网络可以包括铜传输电缆、光纤传输、无线传输、路由器、防火墙、交换机、网关计算机和/或边缘服务器。每个计算/处理设备中的网络适配卡或者网络接口从网络接收计算机可读程序指令,并转发该计算机可读程序指令,以供存储在各个计算/处理设备中的计算机可读存储介质中。

[0061] 用于执行本公开操作的计算机程序指令可以是汇编指令、指令集架构(ISA)指令、机器指令、机器相关指令、微代码、固件指令、状态设置数据、或者以一种或多种编程语言的任意组合编写的源代码或目标代码,所述编程语言包括面向对象的编程语言—诸如Smalltalk、C++等,以及常规的过程式编程语言—诸如“C”语言或类似的编程语言。计算机可读程序指令可以完全地在用户计算机上执行、部分地在用户计算机上执行、作为一个独立的软件包执行、部分在用户计算机上部分在远程计算机上执行、或者完全在远程计算机或服务器上执行。在涉及远程计算机的情形中,远程计算机可以通过任意种类的网络—包括局域网(LAN)或广域网(WAN)—连接到用户计算机,或者,可以连接到外部计算机(例如利用因特网服务提供商来通过因特网连接)。在一些实施例中,通过利用计算机可读程序指令

的状态信息来个性化定制电子电路，例如可编程逻辑电路、现场可编程门阵列(FPGA)或可编程逻辑阵列(PLA)，该电子电路可以执行计算机可读程序指令，从而实现本公开的各个方面。

[0062] 这里参照根据本公开实施例的方法、装置(系统)和计算机程序产品的流程图和/或框图描述了本公开的各个方面。应当理解，流程图和/或框图的每个方框以及流程图和/或框图中各方框的组合，都可以由计算机可读程序指令实现。

[0063] 这些计算机可读程序指令可以提供给通用计算机、专用计算机或其它可编程数据处理装置的处理单元，从而生产出一种机器，使得这些指令在通过计算机或其它可编程数据处理装置的处理单元执行时，产生了实现流程图和/或框图中的一个或多个方框中规定的功能/动作的装置。也可以把这些计算机可读程序指令存储在计算机可读存储介质中，这些指令使得计算机、可编程数据处理装置和/或其他设备以特定方式工作，从而，存储有指令的计算机可读介质则包括一个制造品，其包括实现流程图和/或框图中的一个或多个方框中规定的功能/动作的各个方面的指令。

[0064] 也可以把计算机可读程序指令加载到计算机、其它可编程数据处理装置、或其它设备上，使得在计算机、其它可编程数据处理装置或其它设备上执行一系列操作步骤，以产生计算机实现的过程，从而使得在计算机、其它可编程数据处理装置、或其它设备上执行的指令实现流程图和/或框图中的一个或多个方框中规定的功能/动作。

[0065] 附图中的流程图和框图显示了根据本公开的多个实施例的系统、方法和计算机程序产品的可能实现的体系架构、功能和操作。在这点上，流程图或框图中的每个方框可以代表一个模块、程序段或指令的一部分，所述模块、程序段或指令的一部分包含一个或多个用于实现规定的逻辑功能的可执行指令。在有些作为替换的实现中，方框中所标注的功能也可以以不同于附图中所标注的顺序发生。例如，两个连续的方框实际上可以基本并行地执行，它们有时也可以按相反的顺序执行，这依所涉及的功能而定。也要注意的是，框图和/或流程图中的每个方框、以及框图和/或流程图中的方框的组合，可以用执行规定的功能或动作的专用的基于硬件的系统来实现，或者可以用专用硬件与计算机指令的组合来实现。

[0066] 以上已经描述了本公开的各实施例，上述说明是示例性的，并非穷尽性的，并且也不限于所披露的各实施例。在不偏离所说明的各实施例的范围和精神的情况下，对于本技术领域的普通技术人员来说许多修改和变更都是显而易见的。本文中所用术语的选择，旨在最好地解释各实施例的原理、实际应用或对市场中的技术的技术改进，或者使本技术领域的其它普通技术人员能理解本文披露的各实施例。

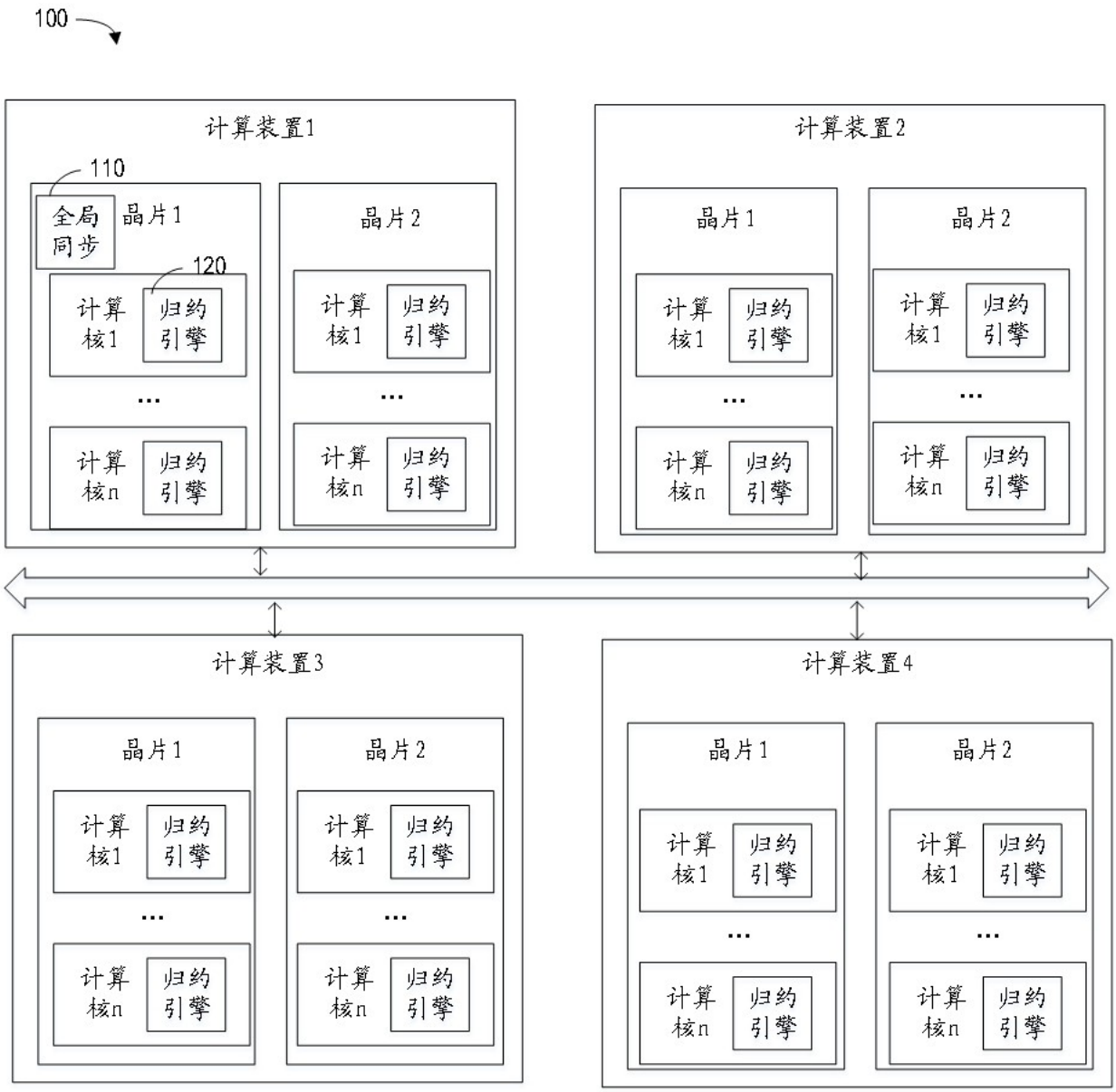

100
计算装置1
110
全局同步
晶片1
120
计算核1
归约引擎
计算核n
晶片2
计算装置2
计算装置3
计算装置4

图1

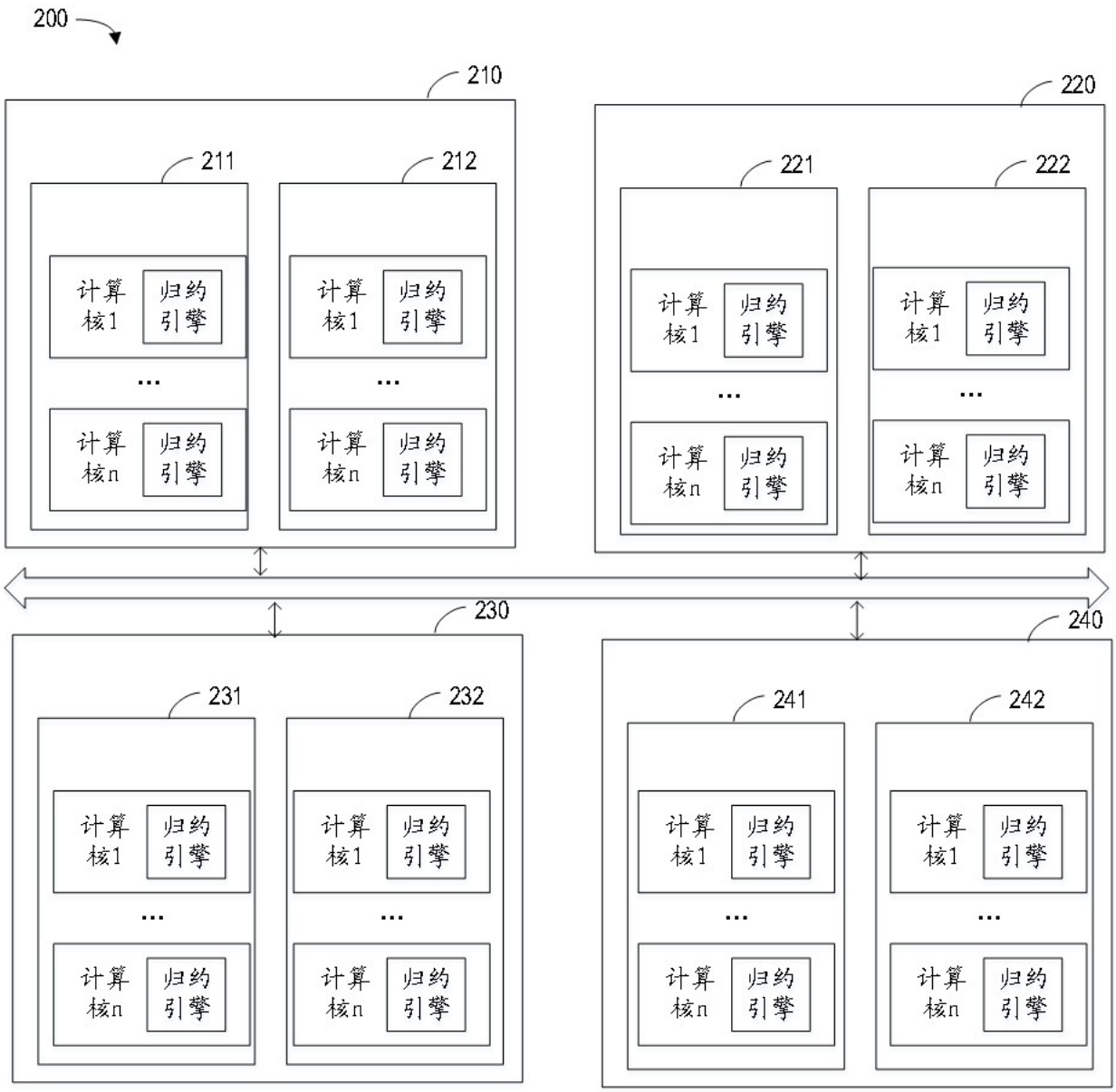

200
210
211
212
220
221
222
230
231
232
240
241
242
计算核1
归约引擎
计算核n
归约引擎
...

图2

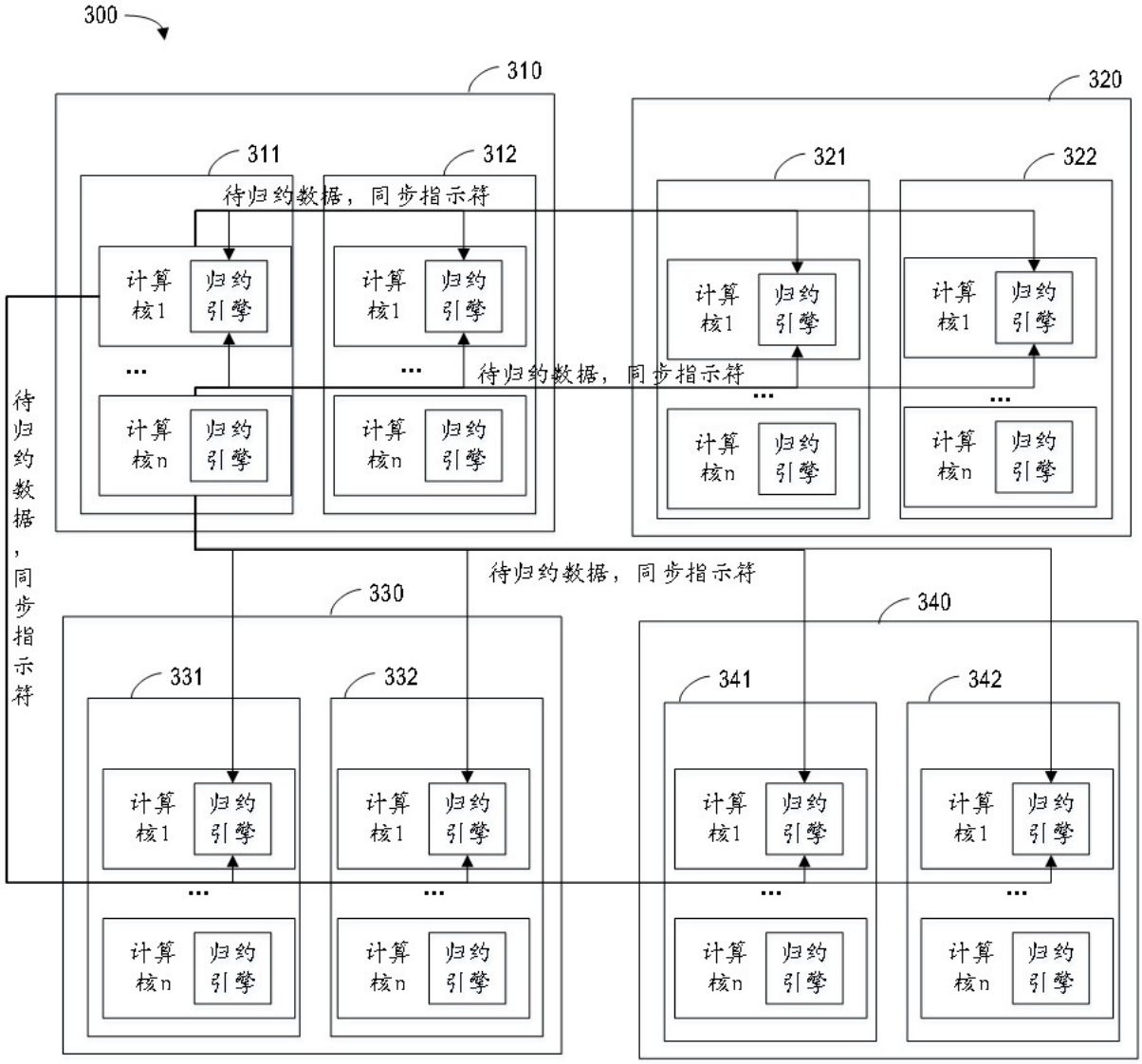

300
310
320
311
312
321
322
待归约数据，同步指示符
计算核1
归约引擎
计算核n
待归约数据，同步指示符
待归约数据，同步指示符
待归约数据，同步指示符
330
340
331
332
341
342

图3

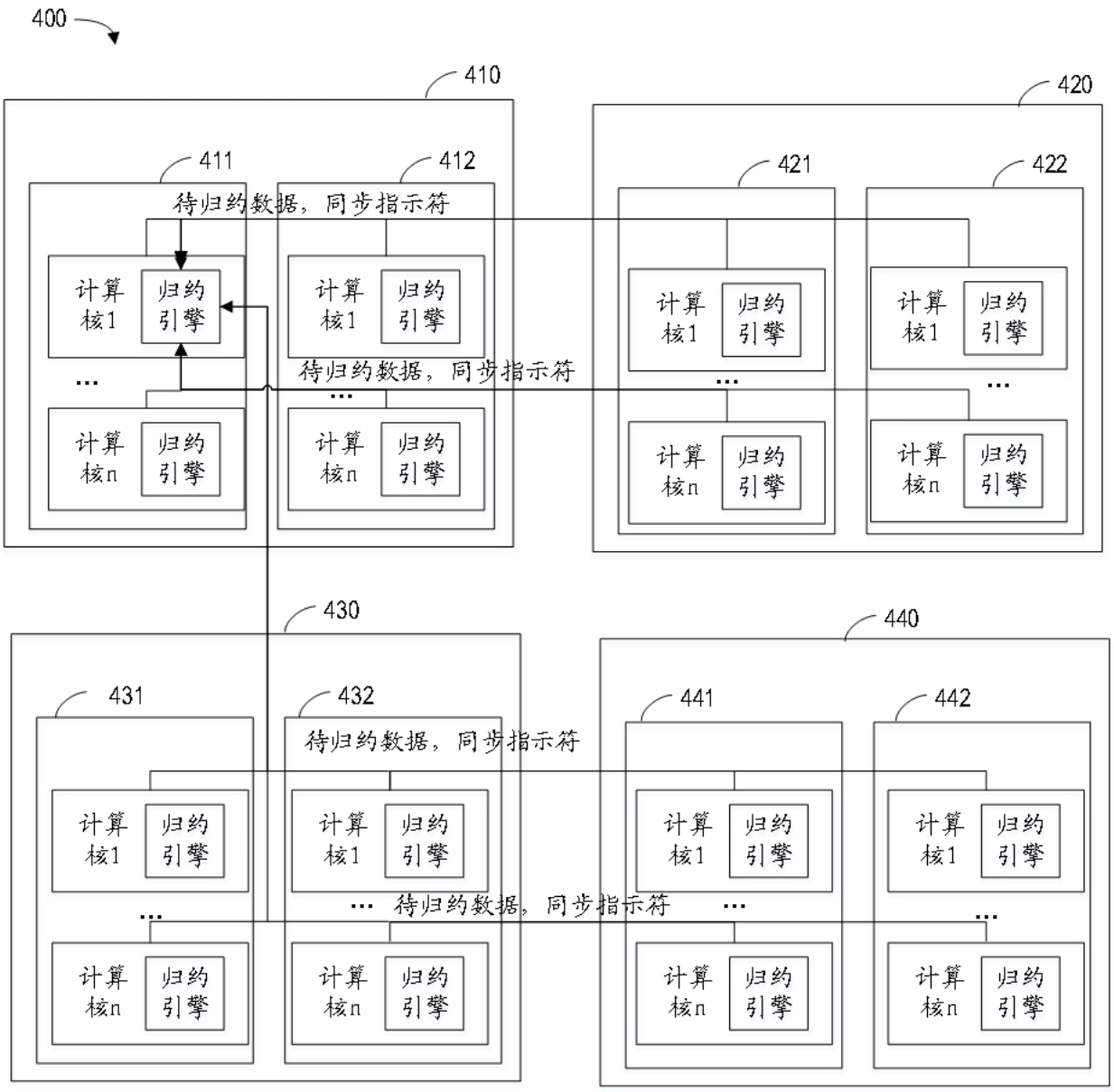

400
410
420
411
412
421
422
待归约数据，同步指示符
计算核1
归约引擎
计算核n
待归约数据，同步指示符
430
440
431
432
441
442
待归约数据，同步指示符
待归约数据，同步指示符

图4

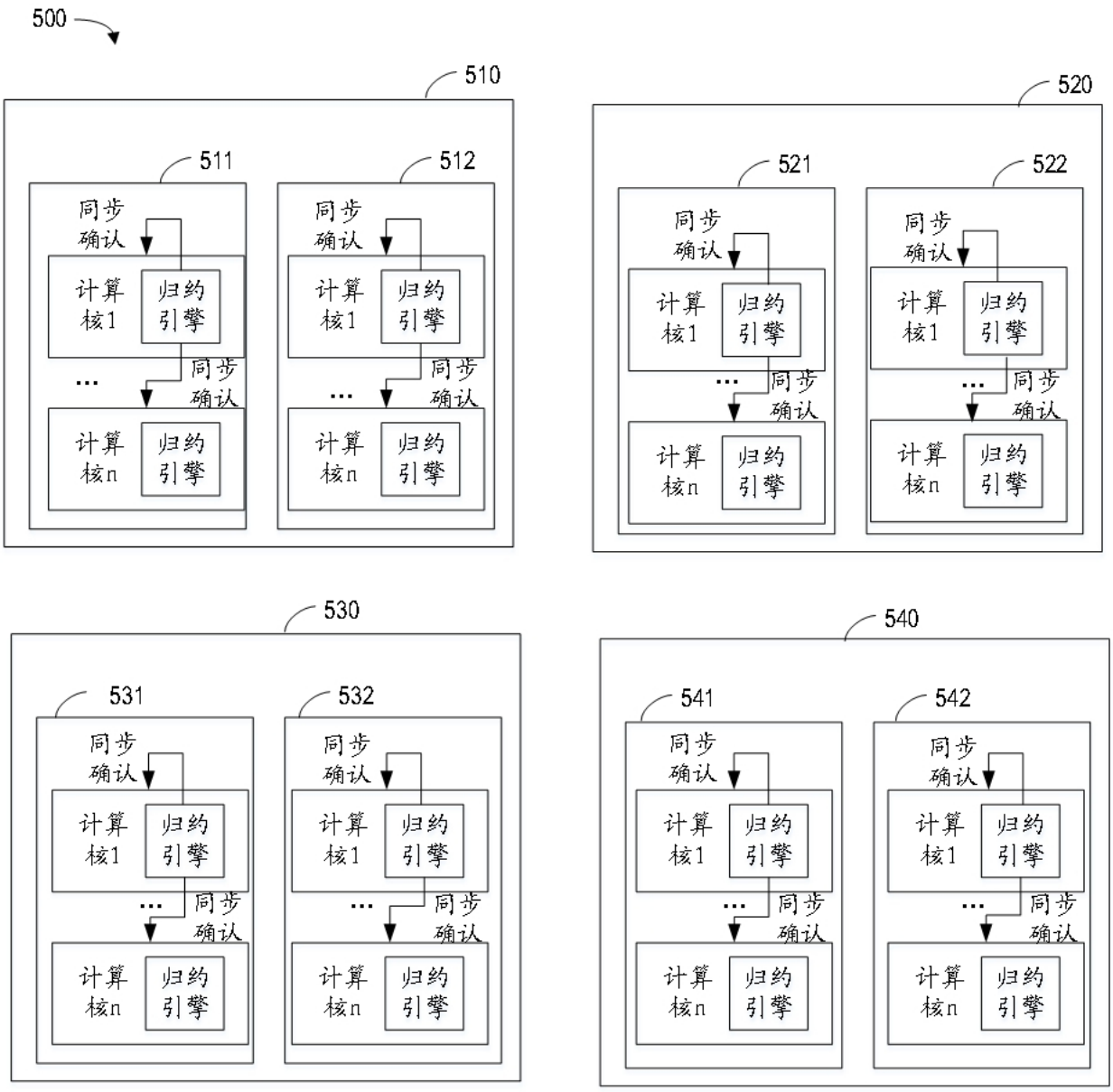

500
510
511
512
520
521
522
530
531
532
540
541
542
同步确认
计算核1
归约引擎
计算核n
...

图5

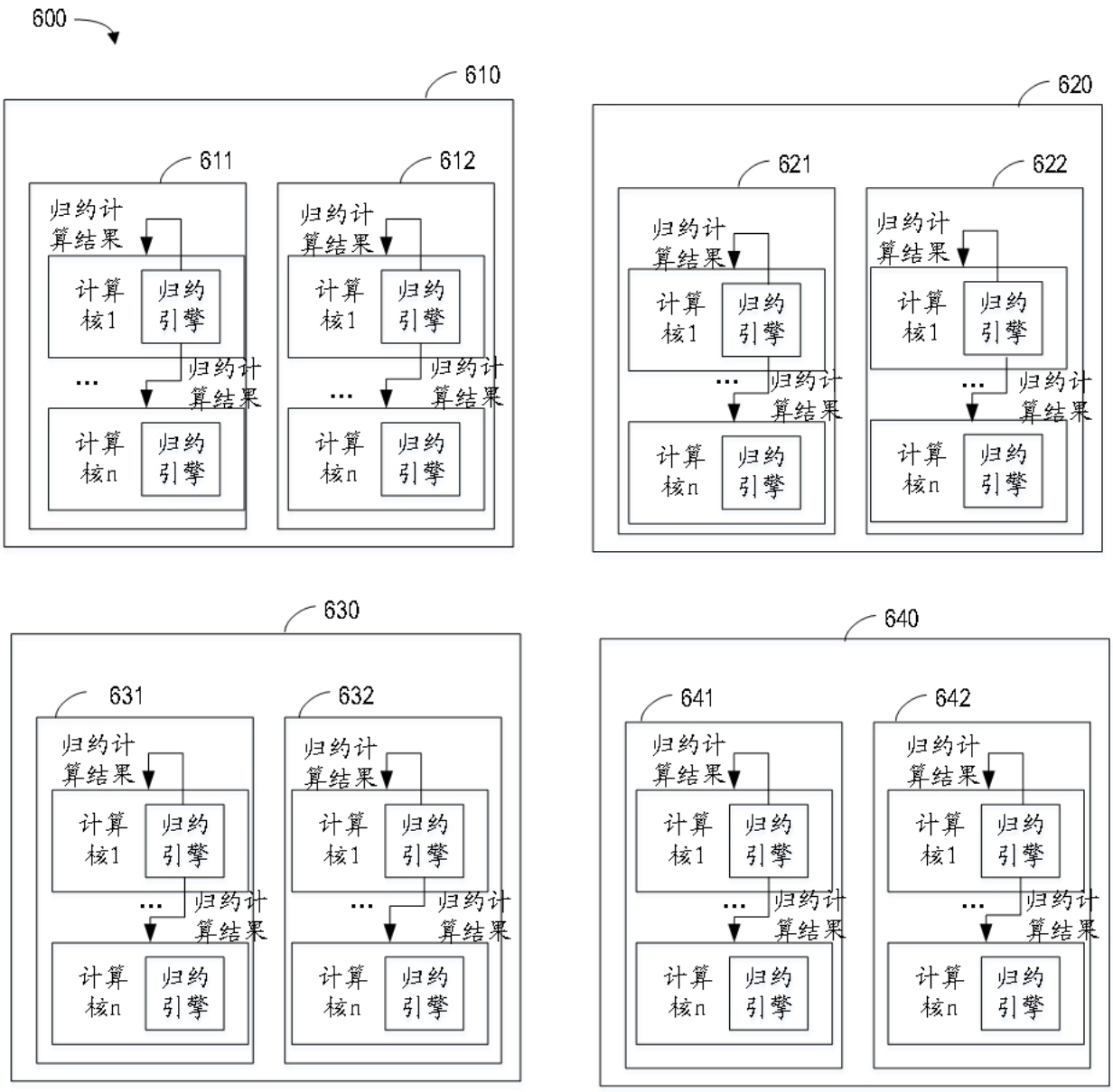

600
610
611
612
归约计算结果
计算核1
归约引擎
...
计算核n
620
621
622
630
631
632
640
641
642

图6

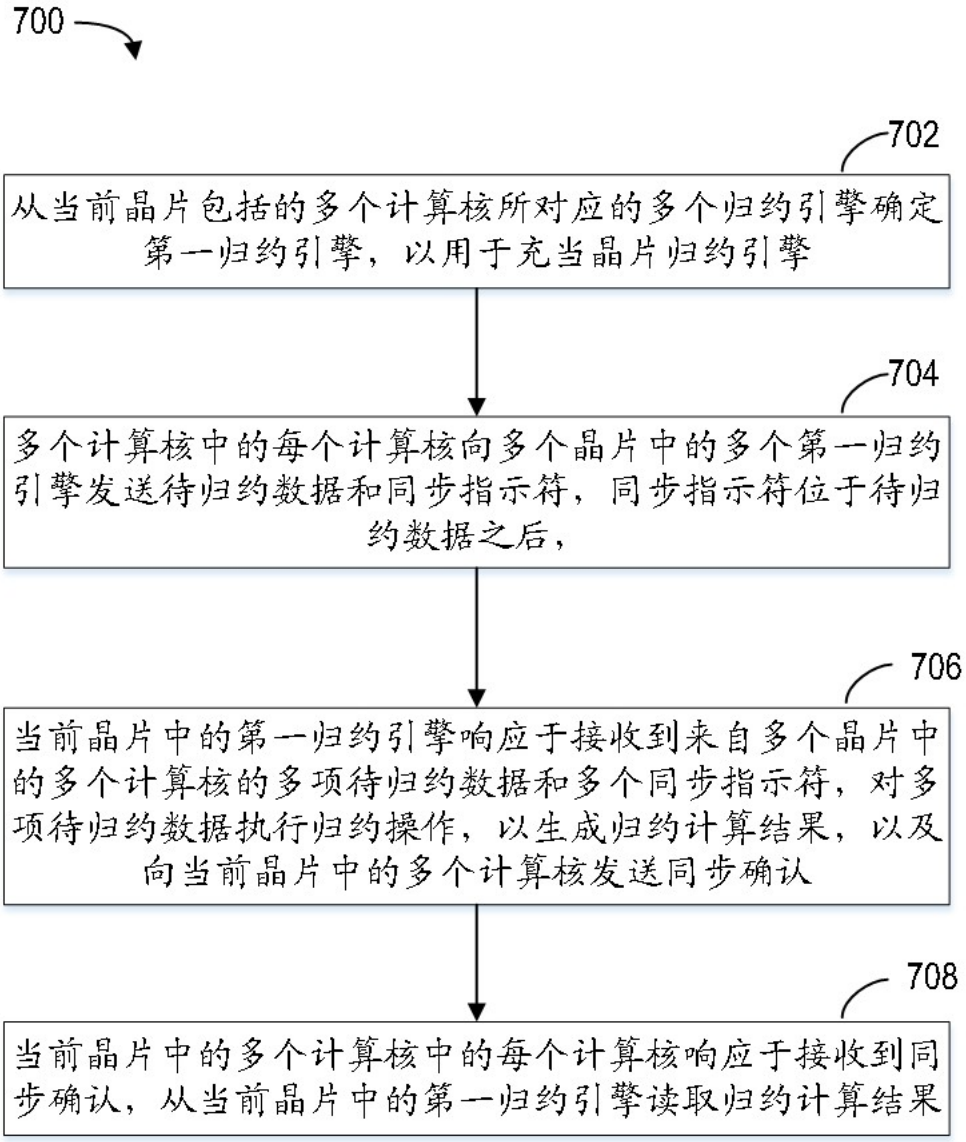

700
702
从当前晶片包括的多个计算核所对应的多个归约引擎确定第一归约引擎，以用于充当晶片归约引擎
704
多个计算核中的每个计算核向多个晶片中的多个第一归约引擎发送待归约数据和同步指示符，同步指示符位于待归约数据之后，
706
当前晶片中的第一归约引擎响应于接收到来自多个晶片中的多个计算核的多项待归约数据和多个同步指示符，对多项待归约数据执行归约操作，以生成归约计算结果，以及向当前晶片中的多个计算核发送同步确认
708
当前晶片中的多个计算核中的每个计算核响应于接收到同步确认，从当前晶片中的第一归约引擎读取归约计算结果

图7