(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 118012631 B
(45)授权公告日 2024.07.05

(21)申请号 202410411948.X

(22)申请日 2024.04.07

(65)同一申请的已公布的文献号
申请公布号 CN 118012631 A

(43)申请公布日 2024.05.10

(73)专利权人 北京壁仞科技开发有限公司
地址 100102 北京市朝阳区京东园四区13号楼-4至33层101内10层201室
专利权人 上海壁仞科技股份有限公司

(72)发明人 请求不公布姓名 请求不公布姓名 请求不公布姓名 请求不公布姓名 请求不公布姓名 请求不公布姓名

(74)专利代理机构 北京同达信恒知识产权代理有限公司 11291
专利代理师 金银花

(51)Int.Cl.
G06F 9/50(2006.01)
G06F 9/54(2006.01)

(56)对比文件
CN 116432718 A,2023.07.14

审查员 白桦

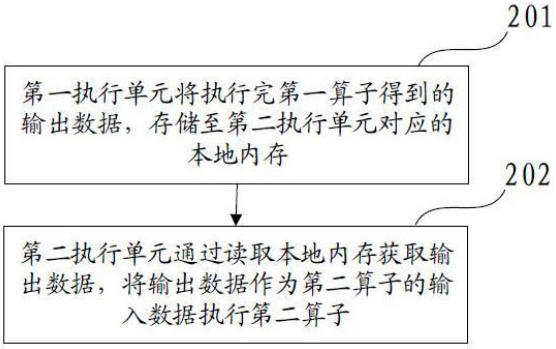
权利要求书2页 说明书9页 附图7页

(54)发明名称

一种算子执行方法、处理设备、存储介质及程序产品

(57)摘要

一种算子执行方法、处理设备、存储介质及程序产品，涉及人工智能技术领域，用以减少算子运算时的访存时延。该算子执行方法适用于具有N个执行单元的处理设备；每个执行单元具有各自对应的本地内存；该方法包括：第一执行单元将执行完第一算子得到的输出数据，存储至第二执行单元对应的本地内存；所述第二执行单元对应的本地内存中具有与第一执行单元之间存在映射关系的地址空间；所述第二执行单元通过读取本地内存获取所述输出数据，将所述输出数据作为第二算子的输入数据执行所述第二算子；所述第二算子为所述第一算子的后一个算子。



CN 118012631 B

1.一种算子执行方法,其特征在于,适用于具有N个执行单元的处理设备;每个执行单元具有各自对应的本地内存;所述N个执行单元被划分为多个执行组;第一执行单元与第二执行单元属于同一个执行组,或,每个执行组中设置有作为核心的执行单元且第一执行单元为各执行组中作为核心的执行单元;所述方法包括:

第一执行单元将执行完第一算子得到的输出数据,存储至第二执行单元对应的本地内存;所述第二执行单元对应的本地内存中具有与第一执行单元之间存在映射关系的地址空间;

所述第二执行单元读取本地内存获取所述输出数据,将所述输出数据作为第二算子的输入数据执行所述第二算子;所述第二算子为所述第一算子的后一个算子。

2.根据权利要求1所述的方法,其特征在于,所述第一执行单元与所述第二执行单元属于同一个执行组;

同一执行组中,每个执行单元的本地内存均具有与其他执行单元之间存在映射关系的地址空间;

任一第一执行单元执行第一算子的输出数据为所述执行组中各执行单元执行第二算子的输入数据;

所述第一执行单元通过统一内存访问的方式将执行完第一算子得到的输出数据,存储至第二执行单元对应的本地内存,包括:

任一第一执行单元将执行完第一算子的输出数据,分别写入所述执行组中各执行单元的本地内存中符合对应映射关系的地址空间中。

3.根据权利要求1所述的方法,其特征在于,所述第一执行单元与所述第二执行单元属于同一个执行组;

每个执行组中设置有作为核心的执行单元;所述第二执行单元为作为核心的执行单元;所述第一执行单元为任一执行单元;

各第一执行单元执行第一算子的输出数据为所述第二执行单元执行第二算子的输入数据;

所述第一执行单元通过统一内存访问的方式将执行完第一算子得到的输出数据,存储至第二执行单元对应的本地内存,包括:

任一第一执行单元将执行完第一算子的输出数据写入所述第二执行单元的本地内存中符合对应映射关系的地址空间中。

4.根据权利要求1所述的方法,其特征在于,所述第一执行单元与所述第二执行单元属于同一个执行组;

每个执行组中设置有作为核心的执行单元;

同一执行组中,每个执行单元的本地内存均具有与作为核心的执行单元之间存在映射关系的地址空间;

所述第一执行单元为作为核心的执行单元,所述第二执行单元为任一执行单元;

所述第一执行单元通过统一内存访问的方式将执行完第一算子得到的输出数据,存储至第二执行单元对应的本地内存,包括:

所述第一执行单元将执行完第一算子的输出数据分别写入各第二执行单元的本地内存中符合对应映射关系的地址空间中。

5.根据权利要求1所述的方法,其特征在于,每个执行组中设置有作为核心的执行单元且所述第一执行单元为各执行组中作为核心的执行单元;

各执行单元均执行第二算子;所述各执行单元的本地内存划分为多个连续存储区域;每个连续存储区域与所述第一执行单元具有地址空间的映射关系;

所述第一执行单元通过统一内存访问的方式将执行完第一算子得到的输出数据,存储至第二执行单元对应的本地内存,包括:

任一第一执行单元将执行完第一算子得到的输出数据,存储至多个第二执行单元对应的连续存储区域中符合对应映射关系的地址空间中。

6.根据权利要求1所述的方法,其特征在于,任一第一执行单元执行完第一算子得到的输出数据为第二执行单元执行第二算子的输入数据;所述第一执行单元与所述第二执行单元为同一执行单元;

所述第一执行单元通过统一内存访问的方式将执行完第一算子得到的输出数据,存储至第二执行单元对应的本地内存,包括:

任一第一执行单元将执行完第一算子的输出数据写入所述第一执行单元的本地内存中。

7.根据权利要求1至6任一项所述的方法,其特征在于,所述方法还包括:

基于芯片上各执行单元的布局、本地内存的访存带宽和各执行单元与本地内存的距离,确定各执行单元所属的执行组以及各执行组中作为核心的执行单元。

8.一种执行算子的处理设备,其特征在于,包括:N个执行单元及每个执行单元对应的本地内存;所述N个执行单元被划分为多个执行组;

任一执行单元用于调用存储的程序指令,按照获得的程序指令执行如权利要求1至7中任一项所述的方法。

9.一种计算机可读存储介质,其特征在于,包括计算机可读指令,当计算机读取并执行所述计算机可读指令时,使得如权利要求1至7中任一项所述的方法实现。

10.一种计算机程序产品,其特征在于,所述计算机程序产品包括存储在计算机可读存储介质上的计算机程序,所述计算机程序包括程序指令,当所述程序指令被计算机设备执行时,使所述计算机设备执行如权利要求1-7任一项所述方法的步骤。

一种算子执行方法、处理设备、存储介质及程序产品

技术领域

[0001] 本申请涉及人工智能技术领域,尤其涉及一种算子执行方法、处理设备、存储介质及程序产品。

背景技术

[0002] 人工智能(artificial intelligence,AI)是通过计算机来模拟人类学习、推理和思考等智能行为的学科,可以应用于图像推理、语音推理等场景。人工智能模型的运算可以由计算图(computation graph)中的算子(operator)来实现。其中,计算图是一种用于表示人工智能模型的计算任务和数据流过程的多图结构。算子是指对人工智能模型中各层的数据所做的各种运算,例如人工智能模型的卷积层对人工智能模型的输入数据所做的卷积运算即为卷积算子。

[0003] 通常,对算子的计算需要多个处理单元协同处理,由于各处理单元以及各处理单元对应的本地内存的排布因素,在处理单元对其它处理单元的数据进行访存时,可能存在较大的时延。对于一些访存密集型场景,在计算过程中会花费大量的时间来进行访存,导致计算时间过长,计算效率低。

[0004] 因此,目前亟需一种方案,用以减少算子运算时的访存时延。

发明内容

[0005] 本申请提供一种算子执行方法及执行算子的处理设备,用以减少算子运算时的访存时延。

[0006] 第一方面,本申请提供一种算子执行方法,该方法适用于具有N个执行单元的处理设备;每个执行单元具有各自对应的本地内存;该方法包括:第一执行单元将执行完第一算子得到的输出数据,存储至第二执行单元对应的本地内存;所述第二执行单元对应的本地内存中具有与第一执行单元之间存在映射关系的地址空间;所述第二执行单元通过读取本地内存获取所述输出数据,将所述输出数据作为第二算子的输入数据执行所述第二算子;所述第二算子为所述第一算子的后一个算子。

[0007] 上述技术方案中,本申请第一执行单元将执行完第一算子得到的输出数据,存储至第二执行单元对应的本地内存,使得第二执行单元执行第二算子时可以从本地内存中读取输入数据,而从本地内存中读取数据相比与从其他执行单元的内存中读取数据速度快很多,因此可以减少算子运算的访存时间,提高计算效率。

[0008] 在一种可能的设计中,所述N个执行单元被划分为多个执行组,所述第一执行单元与所述第二执行单元属于同一个执行组;同一执行组中,每个执行单元的本地内存均具有与其他执行单元之间存在映射关系的地址空间;任一第一执行单元执行第一算子的输出数据为所述执行组中各执行单元执行第二算子的输入数据;所述第一执行单元将执行完第一算子得到的输出数据,存储至第二执行单元对应的本地内存,包括:任一第一执行单元将执行完第一算子的输出数据,分别写入所述执行组中各执行单元的本地内存中符合对应映射

关系的地址空间中。

[0009] 在一种可能的设计中,所述N个执行单元被划分为多个执行组,所述第一执行单元与所述第二执行单元属于同一个执行组;每个执行组中设置有作为核心的执行单元;所述第二执行单元为作为核心的执行单元;所述第一执行单元为任一执行单元;各第一执行单元执行第一算子的输出数据为所述第二执行单元执行第二算子的输入数据;所述第一执行单元将执行完第一算子得到的输出数据,存储至第二执行单元对应的本地内存,包括:任一第一执行单元将执行完第一算子的输出数据写入所述第二执行单元的本地内存中符合对应映射关系的地址空间中。

[0010] 在一种可能的设计中,所述N个执行单元被划分为多个执行组,所述第一执行单元与所述第二执行单元属于同一个执行组;每个执行组中设置有作为核心的执行单元;同一执行组中,每个执行单元的本地内存均具有与作为核心的执行单元之间存在映射关系的地址空间;所述第一执行单元为作为核心的执行单元,所述第二执行单元为任一执行单元;所述第一执行单元将执行完第一算子得到的输出数据,存储至第二执行单元对应的本地内存,包括:所述第一执行单元将执行完第一算子的输出数据分别写入各第二执行单元的本地内存中符合对应映射关系的地址空间中。

[0011] 在一种可能的设计中,所述N个执行单元被划分为多个执行组,每个执行组中设置有作为核心的执行单元;所述第一执行单元为各执行组中作为核心的执行单元;各执行单元均执行第二算子;所述各执行单元的本地内存划分为多个连续存储区域;每个连续存储区域与所述第一执行单元具有地址空间的映射关系;所述第一执行单元将执行完第一算子得到的输出数据,存储至第二执行单元对应的本地内存,包括:任一第一执行单元将执行完第一算子得到的输出数据,存储至多个第二执行单元对应的连续存储区域中符合对应映射关系的地址空间中。

[0012] 在一种可能的设计中,任一第一执行单元执行完第一算子得到的输出数据为第二执行单元执行第二算子的输入数据;所述第一执行单元与所述第二执行单元为同一执行单元;所述第一执行单元将执行完第一算子得到的输出数据,存储至第二执行单元对应的本地内存,包括:任一第一执行单元将执行完第一算子的输出数据写入所述第一执行单元的本地内存中。

[0013] 在一种可能的设计中,所述方法还包括:基于芯片上各执行单元的布局、本地内存的访存带宽和各执行单元与本地内存的距离,确定各执行单元所属的执行组以及各执行组中作为核心的执行单元。

[0014] 第二方面,本申请实施例提供一种执行算子的处理设备,包括:N个执行单元及每个执行单元对应的本地内存;所述N个执行单元被划分为多个执行组;任一执行单元用于调用存储的程序指令,按照获得的程序指令执行如第一方面的任一种可能的设计中所述的方法。

[0015] 第三方面,本申请实施例提供一种计算机可读存储介质,其中存储有计算机可读指令,当计算机读取并执行所述计算机可读指令时,使得上述第一方面的任一种可能的设计中所述的方法实现。

[0016] 第四方面,本申请实施例提供一种计算机程序产品,所述计算机程序产品包括存储在计算机可读存储介质上的计算机程序,所述计算机程序包括程序指令,当所述程序指

令被计算机设备执行时,使所述计算机设备执行上述第一方面的任一种可能的设计中的步骤。

附图说明

[0017] 为了更清楚地说明本申请实施例中的技术方案,下面将对实施例描述中所需要使用的附图作简要介绍,显而易见地,下面描述中的附图仅仅是本申请的一些实施例,对于本领域的普通技术人员来讲,在不付出创造性劳动的前提下,还可以根据这些附图获得其他的附图。

[0018] 图1为本申请实施例提供的一种处理设备的结构示意图;

[0019] 图2为本申请实施例提供的一种算子执行方法的流程示意图;

[0020] 图3为本申请实施例提供的默认模式的示意图;

[0021] 图4为本申请实施例提供的一种执行组的示意图;

[0022] 图5为本申请实施例提供的模式一的示意图;

[0023] 图6为本申请实施例提供的模式二的示意图;

[0024] 图7为本申请实施例提供的模式三的示意图;

[0025] 图8为本申请实施例提供的模式四的示意图。

具体实施方式

[0026] 为了使本申请的目的、技术方案和优点更加清楚,下面将结合附图对本申请作进一步地详细描述,显然,所描述的实施例仅仅是本申请一部分实施例,而不是全部的实施例。基于本申请中的实施例,本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其它实施例,都属于本申请保护的范围。

[0027] 在本申请的实施例中,多个是指两个或两个以上。“第一”、“第二”等词汇,仅用于区分描述的目的,而不能理解为指示或暗示相对重要性,也不能理解为指示或暗示顺序。

[0028] 图1示例性地示出了本申请实施例提供的一种处理设备的结构示意图,该处理设备包括N个执行单元(包括执行单元101-1至101-N)和显存102。

[0029] 每个执行单元可以用于进行算子的运算。显存102可以是高带宽存储器(high bandwidth memory,HBM),也可以是其他类型的存储器。执行单元通过总线103对显存102进行访存。每个执行单元在显存102中具有各自对应的本地内存,参考图1,显存102中为执行单元101-1分配的存储空间102-1作为执行单元101-1的本地内存,为执行单元101-N分配的存储空间102-N作为执行单元101-N的本地内存。每个执行单元对应的本地内存可以存储本执行单元进行算子运算的输出数据,也可以存储其它执行单元进行算子运算的输出数据。并且,当前算子运算的输出数据可以作为下一算子运算的输入数据。

[0030] 本申请中的处理设备除了包含上述结构之外,还可以包括其他结构,对此,本申请不做具体限定。

[0031] 图2示例性地示出了本申请实施例提供的一种算子执行方法的流程示意图,该方法适用于具有N个执行单元的处理设备;每个执行单元具有各自对应的本地内存,如图2所示,该方法包括以下步骤:

[0032] 步骤201、第一执行单元将执行完第一算子得到的输出数据,存储至第二执行单元

对应的本地内存。

[0033] 算子是指对人工智能模型中各层的数据所做的各种运算,算子的类型包括但不限于:卷积算子、全连接算子、池化算子、批归一化BatchNorm算子、索贝尔Sobel算子、重塑Reshape算子、转置Transpose算子等。通过对人工智能模型的运算对应的计算图进行分析可以得到人工智能模型的多个算子组合。

[0034] 一个算子可以由芯片上的一个或多个执行单元计算完成,具体参与计算的执行单元的数目可以根据算子的种类确定。第一执行单元是指执行第一算子的执行单元,第一执行单元可以是一个或多个执行单元;第二执行单元是指执行第二算子的执行单元,第二执行单元可以是一个或多个执行单元。其中,第二执行单元对应的本地内存中具有与第一执行单元之间存在映射关系的地址空间。第二算子为第一算子的后一个算子,第一执行单元将执行完第一算子得到的输出数据,需要在第二执行单元执行第二算子时使用,即为第二执行单元执行第二算子时的输入数据,因此,第一执行单元将执行完第一算子得到的输出数据,存储至第二执行单元对应的本地内存。

[0035] 步骤202、第二执行单元通过读取本地内存获取输出数据,将输出数据作为第二算子的输入数据执行第二算子。

[0036] 执行单元在将数据写入本地内存时是通过统一内存访问(uniform memory access,UMA)的方式来写入的。统一内存访问是指多个执行单元通过同一根总线来访问共享内存。执行单元在从本地内存中读取数据是通过非统一内存访问(non-uniform memory access,NUMA)的方式来读取的。非统一内存访问是指多个执行单元可以同时并行访问各自的内存。非统一内存访问中执行单元在访问本地内存时速度最快,访问的内存距离越远时速度越慢。本申请第一执行单元将执行完第一算子得到的输出数据,存储至第二执行单元对应的本地内存,使得第二执行单元执行第二算子时可以从本地内存中读取输入数据,减少了访存时间。

[0037] 上述算子执行方法在具体实施时包括以下几种模式:

[0038] (一)、默认模式

[0039] 任一第一执行单元执行完第一算子得到的输出数据为第二执行单元执行第二算子的输入数据,其中,第一执行单元与第二执行单元为同一执行单元;在默认模式下,任一第一执行单元将执行完第一算子的输出数据,写入该执行单元的本地内存中。最终,默认模式下每个执行单元的本地内存中均存储有该执行单元执行完第一算子的输出数据。

[0040] 参考图3,以执行单元0、2、8、10为例,执行单元0将执行完第一算子的输出数据0,输出数据0可以是张量(tensor),写入执行单元0的本地内存中。执行单元2将执行完第一算子的输出数据2,写入执行单元2的本地内存中。执行单元8和执行单元10也类似,最终,默认模式下执行单元0、2、8、10的本地内存中均存储有各自执行完第一算子的输出数据。执行单元0、2、8、10在执行第二算子时,通过读取本地内存获取各自执行完第一算子的输出数据,作为第二算子的输入数据执行第二算子。

[0041] 类似地,默认模式下其它参与第一算子运算的各执行单元执行完第一算子后,也是将各自执行完第一算子的输出数据写入各自的本地内存中。

[0042] 上述默认模式适用于,任一执行单元在执行第二算子时需要使用该执行单元执行第一算子后的输出数据。此种情况下,各执行单元将执行完第一算子得到的输出数据,分别

存储至各自的本地内存,使得各执行单元在执行第二算子时可以从各自本地内存中读取输入数据。

[0043] (二)、模式一

[0044] 模式一下N个执行单元被划分为多个执行组,第一执行单元与第二执行单元属于同一个执行组。具体地,可以基于芯片上各执行单元的布局、本地内存的访存带宽和各执行单元与本地内存的距离,确定各执行单元所属的执行组,使得整体带宽访存效率和访存时延最优。

[0045] 示例性地,图4为一种执行组的划分方式,图4所示的芯片上设置有16个执行单元,图4中每个圆圈代表一个执行单元,16个执行单元在芯片上的布局如图所示,对于此种布局方式,可以将16个执行单元划分为4个执行组,参考图4,第一执行组包括执行单元0、2、8、10;第二执行组包括执行单元1、3、9、11;第三执行组包括执行单元4、6、12、14;第四执行组包括执行单元5、7、13、15。

[0046] 需要说明的是,上述只是对16个执行单元划分执行组的其中一种划分方式,具体执行组的数目以及每个执行组中包括哪几个执行单元,可以根据实际芯片上各执行单元的布局、本地内存的访存带宽和各执行单元与本地内存的距离来确定。

[0047] 在模式一下,同一执行组中每个执行单元的本地内存均具有与其他执行单元之间存在映射关系的地址空间。任一第一执行单元执行第一算子的输出数据为执行组中各执行单元执行第二算子的输入数据,任一第一执行单元将执行完第一算子的输出数据,分别写入执行组中各执行单元的本地内存中符合对应映射关系的地址空间中。最终,模式一下一个执行组的每个执行单元的本地内存中均存储有各执行单元执行完第一算子的全量输出数据。

[0048] 以第一执行组为例,参考图5,执行单元0、2、8、10的本地内存均具有与执行单元0、2、8、10存在映射关系的地址空间,也就是说,执行单元0、2、8、10的本地内存上均分配了存储执行单元0、2、8、10执行第一算子的输出数据的地址空间。执行单元0将执行完第一算子的输出数据0,分别写入执行单元0、2、8、10的本地内存中为执行单元0的输出数据0分配的地址空间中。执行单元2将执行完第一算子的输出数据2,分别写入执行单元0、2、8、10的本地内存中为执行单元2的输出数据2分配的地址空间中。执行单元8和执行单元10也类似,最终,模式一下执行单元0、2、8、10的本地内存中均存储有各执行单元执行完第一算子的全量输出数据(输出数据0、输出数据2、输出数据8和输出数据10)。执行单元0、2、8、10在执行第二算子时,通过读取本地内存获取各执行单元执行完第一算子的全量输出数据(输出数据0、输出数据2、输出数据8和输出数据10),作为第二算子的输入数据执行第二算子。

[0049] 类似地,模式一下第二执行组中执行单元1、3、9、11将各自执行完第一算子的输出数据1、3、9、11,分别写入执行单元1、3、9、11的本地内存中为执行单元1、3、9、11的输出数据1、3、9、11分配的地址空间中。其它执行组也类似,在此不做赘述。

[0050] 具体地,每个执行单元均设置有对应的地址偏移,任一执行单元执行完第一算子得到输出数据后,将执行组内每个执行单元对应的本地内存的基地址与该执行单元对应的地址偏移相加,得到该执行单元的输出数据在各执行单元对应的本地内存中应存储的地址,然后该执行单元将执行第一算子的输出数据分别写入对应的地址空间中。

[0051] 上述模式一适用于,对于任一执行组,第一算子由执行组的各执行单元来执行计

算,各执行单元执行第一算子后各执行单元得到各自的输出数据,且第二算子也由执行组的各执行单元执行计算,执行组的各执行单元在执行第二算子时需要使用各执行单元执行第一算子后各执行单元的全量输出数据。此种情况下,各执行单元将执行完第一算子得到的输出数据,分别存储至各计算单元对应的本地内存,使得各执行单元在执行第二算子时可以从本地内存中读取输入数据(各执行单元执行完第一算子的全量输出数据),从本地内存中读取数据速度快,因此可以减少算子运算的访存时间,提高计算效率。

[0052] (三)、模式二

[0053] 模式二下N个执行单元被划分为多个执行组,第一执行单元与第二执行单元属于同一个执行组。具体执行组的划分方式参考模式一,本申请再此不做赘述。在模式二下每个执行组中设置有作为核心的执行单元,核心的执行单元作为执行组的枢纽,可以执行对执行组中各执行单元的输出数据进行汇总类的运算,例如规约算子的计算可以在作为核心的执行单元上执行。具体地,可以基于芯片上各执行单元的布局、本地内存的访存带宽和各执行单元与本地内存的距离,确定执行组中作为核心的执行单元,使得整体带宽访存效率和访存时延最优。

[0054] 示例性地,参考图4,第一执行组中可以设置执行单元10作为执行单元0、2、8、10核心的执行单元;第二执行组中可以设置执行单元9作为执行单元1、3、9、11核心的执行单元;第三执行组中可以设置执行单元6作为执行单元4、6、12、14核心的执行单元;第四执行组中可以设置执行单元5作为执行单元5、7、13、15核心的执行单元。

[0055] 在模式二下,第二执行单元为执行组中作为核心的执行单元,第一执行单元为执行组任一执行单元,各第一执行单元执行第一算子的输出数据为第二执行单元执行第二算子的输入数据。任一第一执行单元将执行完第一算子的输出数据写入第二执行单元的本地内存中符合对应映射关系的地址空间中。最终,模式二下一个执行组中作为核心的执行单元的本地内存中存储有各执行单元执行完第一算子的全量输出数据。

[0056] 以第一执行组为例,参考图6,第一执行组中设置执行单元10作为核心的执行单元,执行单元10的本地内存具有与执行单元0、2、8、10存在映射关系的地址空间,也就是说,执行单元10的本地内存上均分配了存储执行单元0、2、8、10执行第一算子的输出数据的地址空间。执行单元0将执行完第一算子的输出数据0,写入执行单元10的本地内存中为执行单元0的输出数据0分配的地址空间中。执行单元2将执行完第一算子的输出数据2,写入执行单元10的本地内存中为执行单元2的输出数据2分配的地址空间中。执行单元8和执行单元10也类似,最终,模式二下执行单元10的本地内存中存储有各执行单元执行完第一算子的全量输出数据(输出数据0、输出数据2、输出数据8和输出数据10)。执行单元10在执行第二算子时,通过读取本地内存获取各执行单元执行完第一算子的全量输出数据(输出数据0、输出数据2、输出数据8和输出数据10),作为第二算子的输入数据执行第二算子。

[0057] 类似地,模式二下第二执行组中执行单元1、3、9、11将各自执行完第一算子的输出数据1、3、9、11,写入执行单元9的本地内存中为执行单元1、3、9、11的输出数据1、3、9、11分配的地址空间中。其它执行组也类似,在此不做赘述。

[0058] 具体地,每个执行单元均设置有对应的地址偏移,任一执行单元执行完第一算子得到输出数据后,将执行组内作为核心的执行单元对应的本地内存的基地址与该执行单元对应的地址偏移相加,得到该执行单元的输出数据在作为核心的执行单元对应的本地内存

中应存储的地址,然后该执行单元将执行第一算子的输出数据分别写入对应的地址空间中。

[0059] 上述模式二适用于,对于任一执行组,第一算子由执行组的各执行单元来执行计算,各执行单元执行第一算子后各执行单元得到各自的输出数据,且第二算子由执行组中作为核心的执行单元执行计算,执行组中作为核心的执行单元在执行第二算子时需要使用各执行单元执行第一算子后各执行单元的全量输出数据。此种情况下,各执行单元将执行完第一算子得到的输出数据,存储至作为核心的计算单元对应的本地内存,使得作为核心的执行单元在执行第二算子时可以从本地内存中读取输入数据(各执行单元执行完第一算子的全量输出数据),从本地内存中读取数据速度快,因此可以减少算子运算的访存时间,提高计算效率。

[0060] (四)、模式三

[0061] 模式三下N个执行单元被划分为多个执行组,第一执行单元与第二执行单元属于同一个执行组,每个执行组中设置有作为核心的执行单元。具体执行组的划分方式参考模式一,执行组中设置的作为核心的执行单元参考模式二,本申请在此不做赘述。

[0062] 在模式三下,同一执行组中,每个执行单元的本地内存均具有与作为核心的执行单元之间存在映射关系的地址空间。第一执行单元为作为核心的执行单元,第二执行单元为任一执行单元。第一执行单元将执行完第一算子的输出数据分别写入各第二执行单元的本地内存中符合对应映射关系的地址空间中。最终,模式三下一个执行组的每个执行单元的本地内存中均存储有作为核心的执行单元执行完第一算子的输出数据。

[0063] 以第一执行组为例,参考图7,第一执行组中设置执行单元10作为核心的执行单元,执行单元0、2、8、10的本地内存均具有与作为核心的执行单元10存在映射关系的地址空间,也就是说,执行单元0、2、8、10的本地内存上均分配了存储执行单元10执行完第一算子的输出数据的地址空间。执行单元10将执行完第一算子的输出数据10,分别写入执行单元0、2、8、10的本地内存上为执行单元10的输出数据10分配的地址空间中。最终,模式三下执行单元0、2、8、10的本地内存中均存储有执行单元10执行完第一算子的输出数据(输出数据10)。执行单元0、2、8、10在执行第二算子时,通过读取本地内存获取执行单元10执行完第一算子的输出数据(输出数据10),作为第二算子的输入数据执行第二算子。

[0064] 类似地,模式三下第二执行组中作为核心的执行单元9将执行完第一算子的输出数据9分别写入执行单元1、3、9、11的本地内存中为执行单元9的输出数据9分配的地址空间中。其它执行组也类似,在此不做赘述。

[0065] 具体地,可以在第一执行单元上开4块相同大小的地址空间,先将输出数据分别写入4块地址空间中,再将每块地址空间中的输出数据映射到相应的第二执行单元的本地内存中。例如,在作为核心的执行单元10上开4块相同大小的地址空间,先将输出数据10分别写入4块地址空间中,再将第一块地址空间中的输出数据10映射到执行单元0的本地内存中;将第二块地址空间中的输出数据10映射到执行单元2的本地内存中;将第三块地址空间中的输出数据10映射到执行单元8的本地内存中;将第四块地址空间中的输出数据10映射到执行单元10的本地内存中。

[0066] 上述模式三适用于,对于任一执行组,第一算子由执行组中作为核心的执行单元来执行计算,作为核心的执行单元执行第一算子后得到输出数据,且第二算子由执行组的

各执行单元执行计算,执行组的各执行单元在执行第二算子时需要使用作为核心的执行单元执行第一算子后的输出数据。此种情况下,作为核心的执行单元将执行完第一算子得到的输出数据,分别存储至各执行单元对应的本地内存,使得各执行单元在执行第二算子时可以从本地内存中读取输入数据(作为核心的执行单元执行完第一算子的输出数据),从本地内存中读取数据速度快,因此可以减少算子运算的访存时间,提高计算效率。

[0067] (五)、模式四

[0068] 模式四下N个执行单元被划分为多个执行组,每个执行组中设置有作为核心的执行单元,第一执行单元为各执行组中作为核心的执行单元。具体执行组的划分方式参考模式一,执行组中设置的作为核心的执行单元参考模式二,本申请再此不做赘述。

[0069] 在模式四下,各执行单元均执行第二算子,各执行单元的本地内存划分为多个连续存储区域,每个连续存储区域与第一执行单元具有地址空间的映射关系,任一第一执行单元将执行完第一算子得到的输出数据,存储至多个第二执行单元对应的连续存储区域中符合对应映射关系的地址空间中。最终,模式四下每个连续存储区域中均存储有各执行组作为核心的执行单元执行完第一算子的全量输出数据。

[0070] 参考图8,连续存储区域一由执行单元0至7的本地内存组成,连续存储区域二由执行单元8至15的本地内存组成。连续存储区域一和连续存储区域二与第一执行单元具有地址空间的映射关系。也就是说,连续存储区域一和连续存储区域二上均分配了存储作为核心的执行单元5、6、9、10执行完第一算子的输出数据的地址空间。作为核心的执行单元5、6、9、10将执行完第一算子的输出数据5、6、9、10分别写入连续存储区域一和连续存储区域二中。最终,连续存储区域一和连续存储区域二中均存储有作为核心的执行单元5、6、9、10执行完第一算子的输出数据(输出数据5、输出数据6、输出数据9和输出数据10)。执行单元0至15中的任一执行单元在执行第二算子时,通过读取临近的连续存储区域(连续存储区域一或连续存储区域二)获取各作为核心的执行单元执行完第一算子的全量输出数据(输出数据5、输出数据6、输出数据9和输出数据10),作为第二算子的输入数据执行第二算子。具体来说,执行单元0至7中的任一执行单元在执行第二算子时,通过读取连续存储区域一来获取各作为核心的执行单元执行完第一算子的全量输出数据,执行单元8至15中的任一执行单元在执行第二算子时,通过读取连续存储区域二来获取各核心的执行单元执行完第一算子的全量输出数据。

[0071] 上述模式四适用于,第一算子由各执行组的作为核心的执行单元来执行计算,各作为核心的执行单元执行第一算子后各作为核心的执行单元得到各自的输出数据,且第二算子由各个执行单元执行计算,各个执行单元在执行第二算子时需要使用各作为核心的执行单元执行第一算子后各作为核心的执行单元的全量输出数据。此种情况下,各作为核心的执行单元将执行完第一算子得到的输出数据,存储至多个第二执行单元对应的连续存储区域中,使得各执行单元在执行第二算子时可以从临近的内存中读取输入数据(各作为核心的执行单元执行完第一算子的全量输出数据),从临近的内存中读取数据相比于从原端内存中读取数据速度更快,因此可以减少算子运算的访存时间,提高计算效率。

[0072] 基于相同的技术构思,本申请实施例提供一种执行算子的处理设备,包括:N个执行单元及每个执行单元对应的本地内存;所述N个执行单元被划分为多个执行组;任一执行单元用于调用存储的程序指令,按照获得的程序指令执行上述任一方式所列的算子执行方

法。

[0073] 基于相同的技术构思,本申请实施例提供一种计算机可读存储介质,其中存储有计算机可读指令,当计算机读取并执行所述计算机可读指令时,使得上述任一方式所列的算子执行方法实现。

[0074] 基于相同的技术构思,本申请实施例还提供一种计算机程序产品,所述计算机程序产品包括存储在计算机可读存储介质上的计算机程序,所述计算机程序包括程序指令,当所述程序指令被计算机设备执行时,使所述计算机设备执行上述任一方式所列的算子执行方法中的步骤。

[0075] 本领域内的技术人员应明白,本申请的实施例可提供为方法、系统、或计算机程序产品。因此,本申请可采用完全硬件实施例、完全软件实施例、或结合软件和硬件方面的实施例的形式。而且,本申请可采用在一个或多个其中包含有计算机可用程序代码的计算机可用存储介质(包括但不限于磁盘存储器、CD-ROM、光学存储器等)上实施的计算机程序产品的形式。

[0076] 本申请是参照根据本申请实施例的方法、设备(系统)、和计算机程序产品的流程图和/或方框图来描述的。应理解可由计算机程序指令实现流程图和/或方框图中的每一流程和/或方框、以及流程图和/或方框图中的流程和/或方框的结合。可提供这些计算机程序指令到通用计算机、专用计算机、嵌入式处理机或其他可编程数据处理设备的处理器以产生一个机器,使得通过计算机或其他可编程数据处理设备的处理器执行的指令产生用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的装置。

[0077] 这些计算机程序指令也可存储在能引导计算机或其他可编程数据处理设备以特定方式工作的计算机可读存储器中,使得存储在该计算机可读存储器中的指令产生包括指令装置的制造品,该指令装置实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能。

[0078] 这些计算机程序指令也可装载到计算机或其他可编程数据处理设备上,使得在计算机或其他可编程设备上执行一系列操作步骤以产生计算机实现的处理,从而在计算机或其他可编程设备上执行的指令提供用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的步骤。

[0079] 尽管已描述了本申请的优选实施例,但本领域内的技术人员一旦得知了基本创造性概念,则可对这些实施例做出另外的变更和修改。所以,所附权利要求意欲解释为包括优选实施例以及落入本申请范围的所有变更和修改。

[0080] 显然,本领域的技术人员可以对本申请进行各种改动和变型而不脱离本申请的精神和范围。这样,倘若本申请的这些修改和变型属于本申请权利要求及其等同技术的范围之内,则本申请也意图包含这些改动和变型在内。

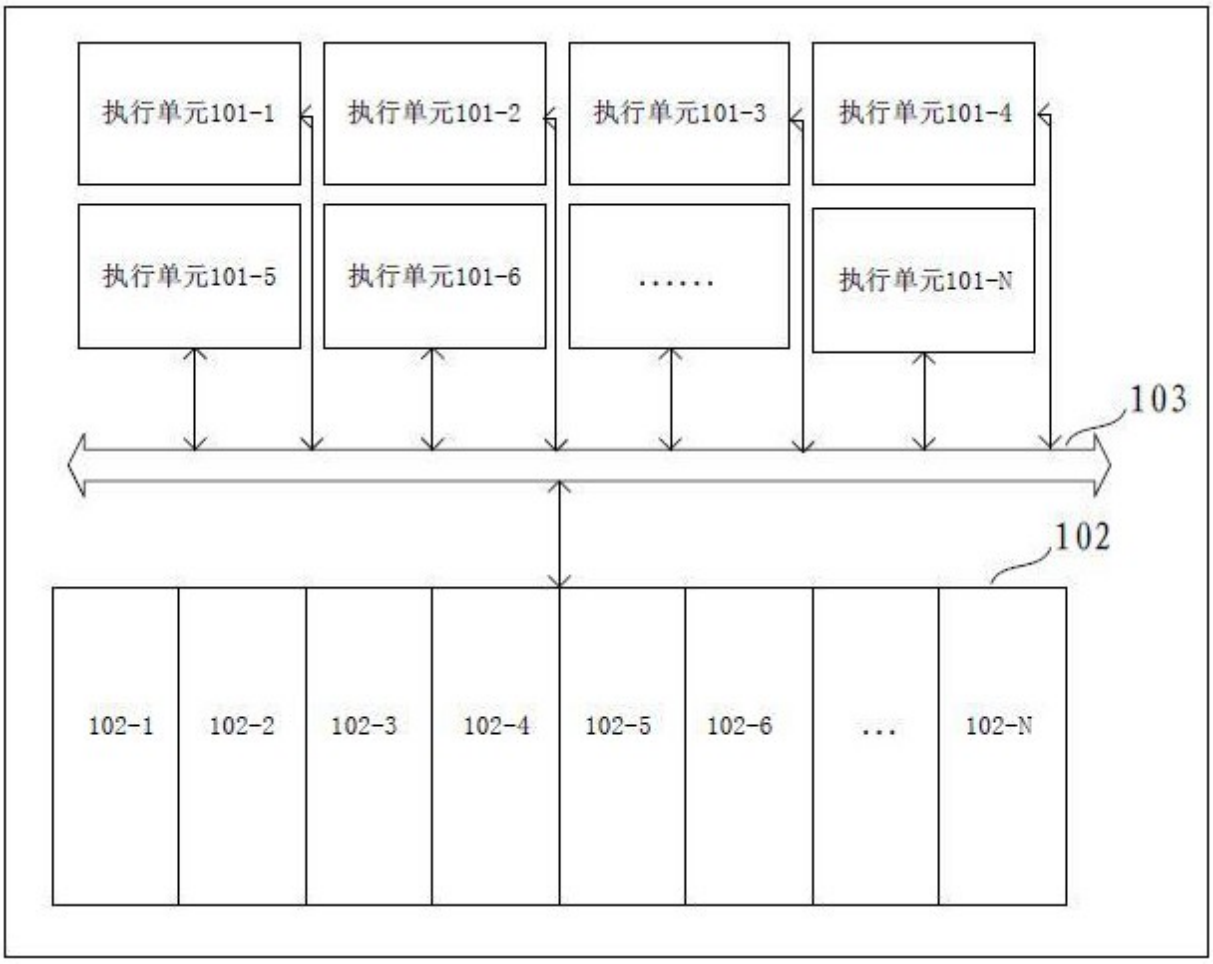

执行单元101-1
执行单元101-2
执行单元101-3
执行单元101-4
执行单元101-5
执行单元101-6
.....
执行单元101-N
103
102
102-1
102-2
102-3
102-4
102-5
102-6
...
102-N

图 1

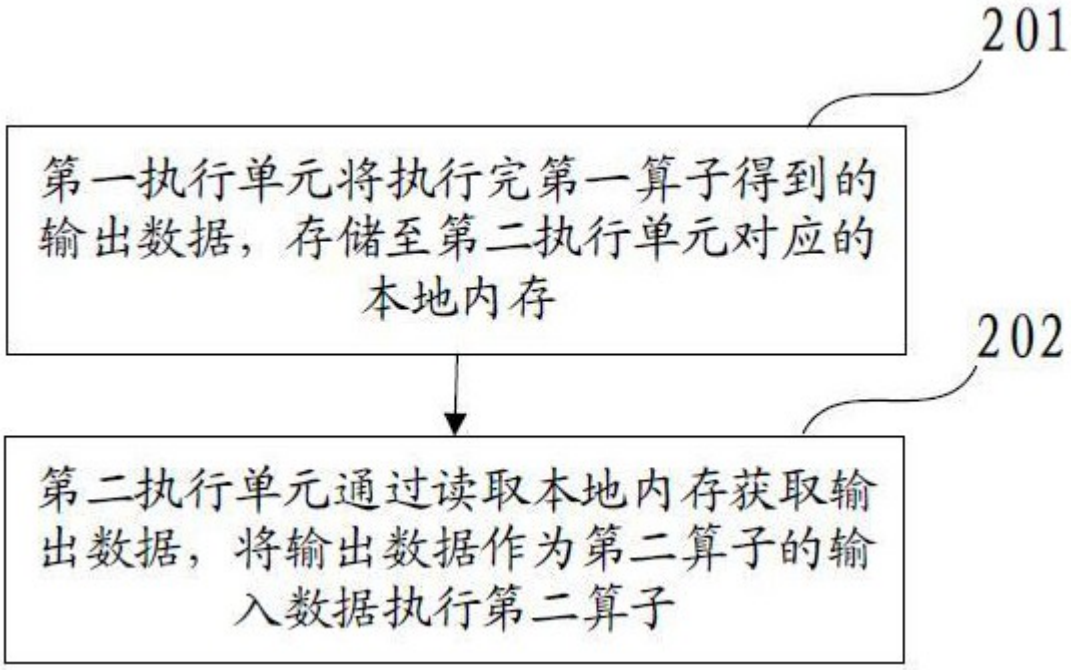

201
第一执行单元将执行完第一算子得到的输出数据，存储至第二执行单元对应的本地内存
202
第二执行单元通过读取本地内存获取输出数据，将输出数据作为第二算子的输入数据执行第二算子

图 2

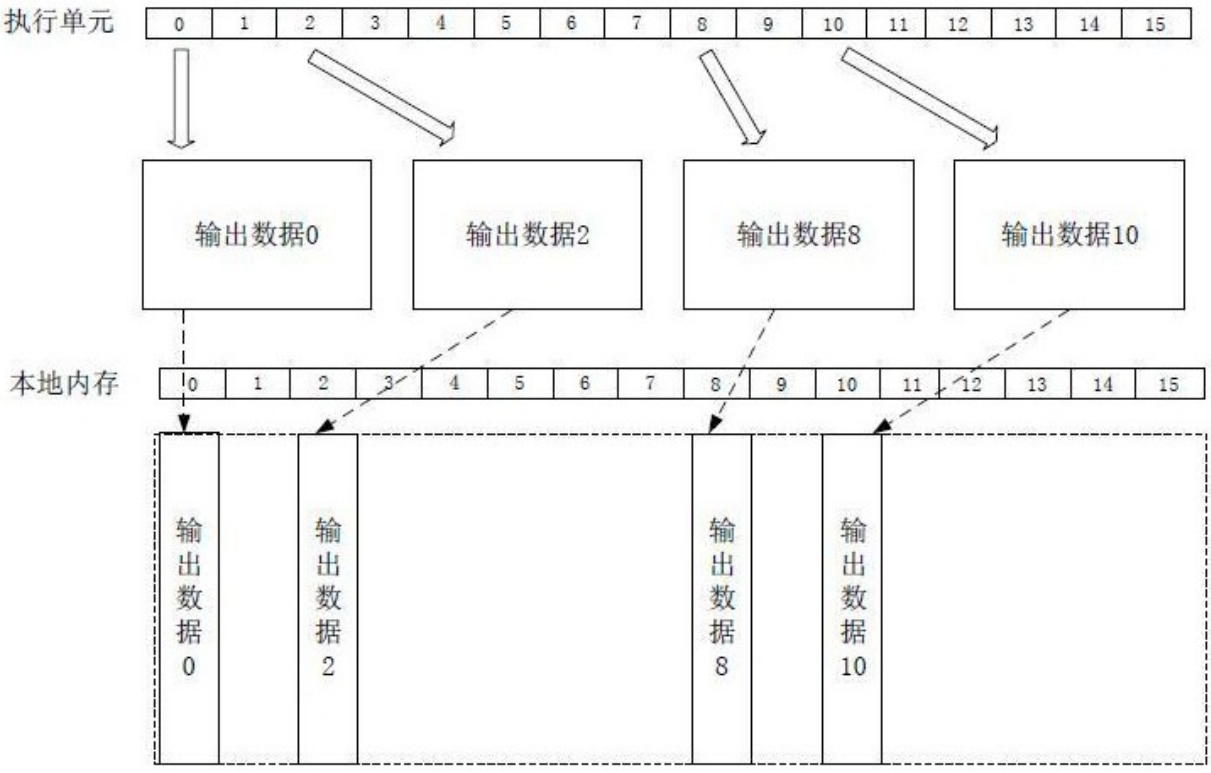

执行单元
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
输出数据0
输出数据2
输出数据8
输出数据10
本地内存
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
输出数据0
输出数据2
输出数据8
输出数据10

图 3

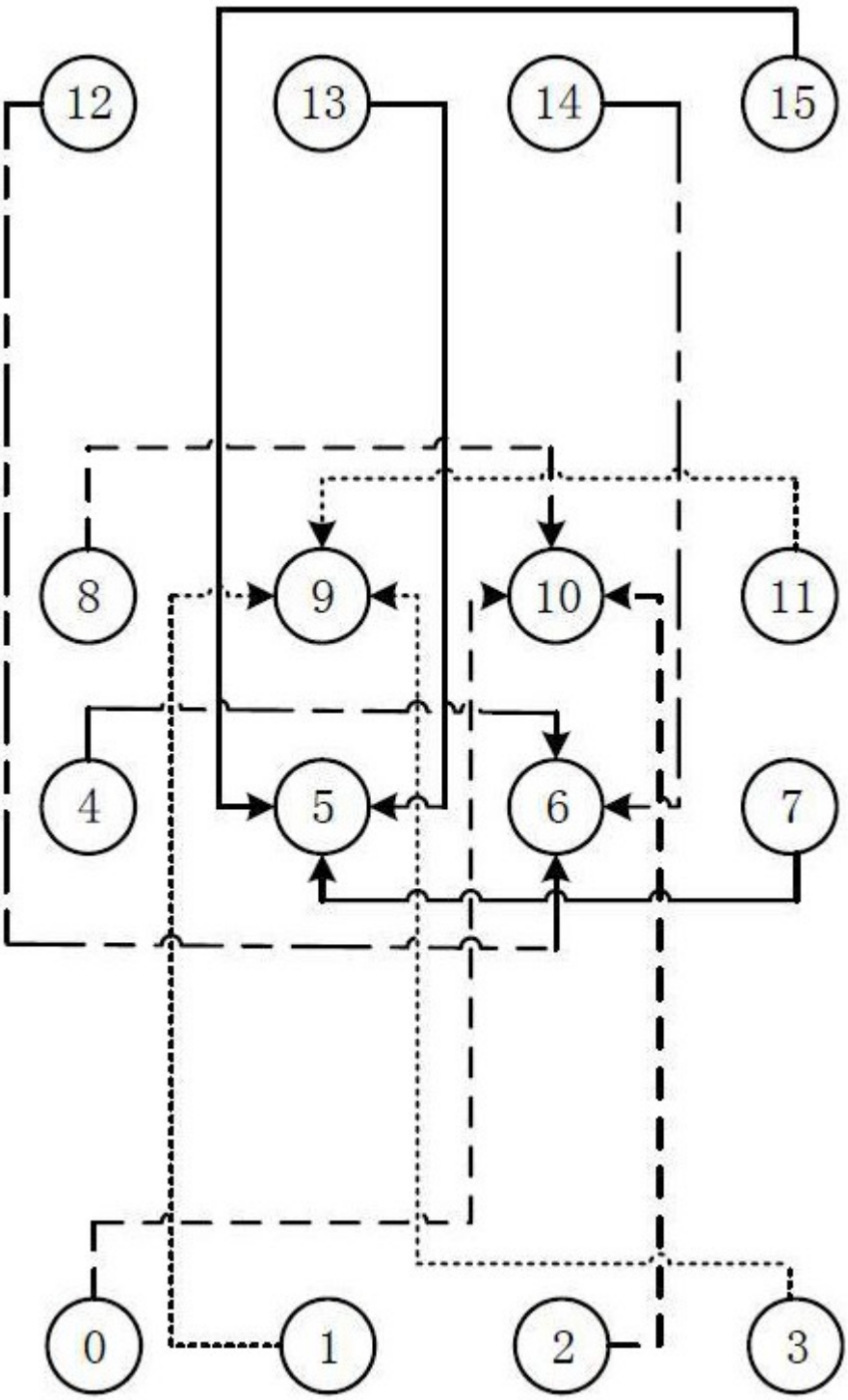


图 4

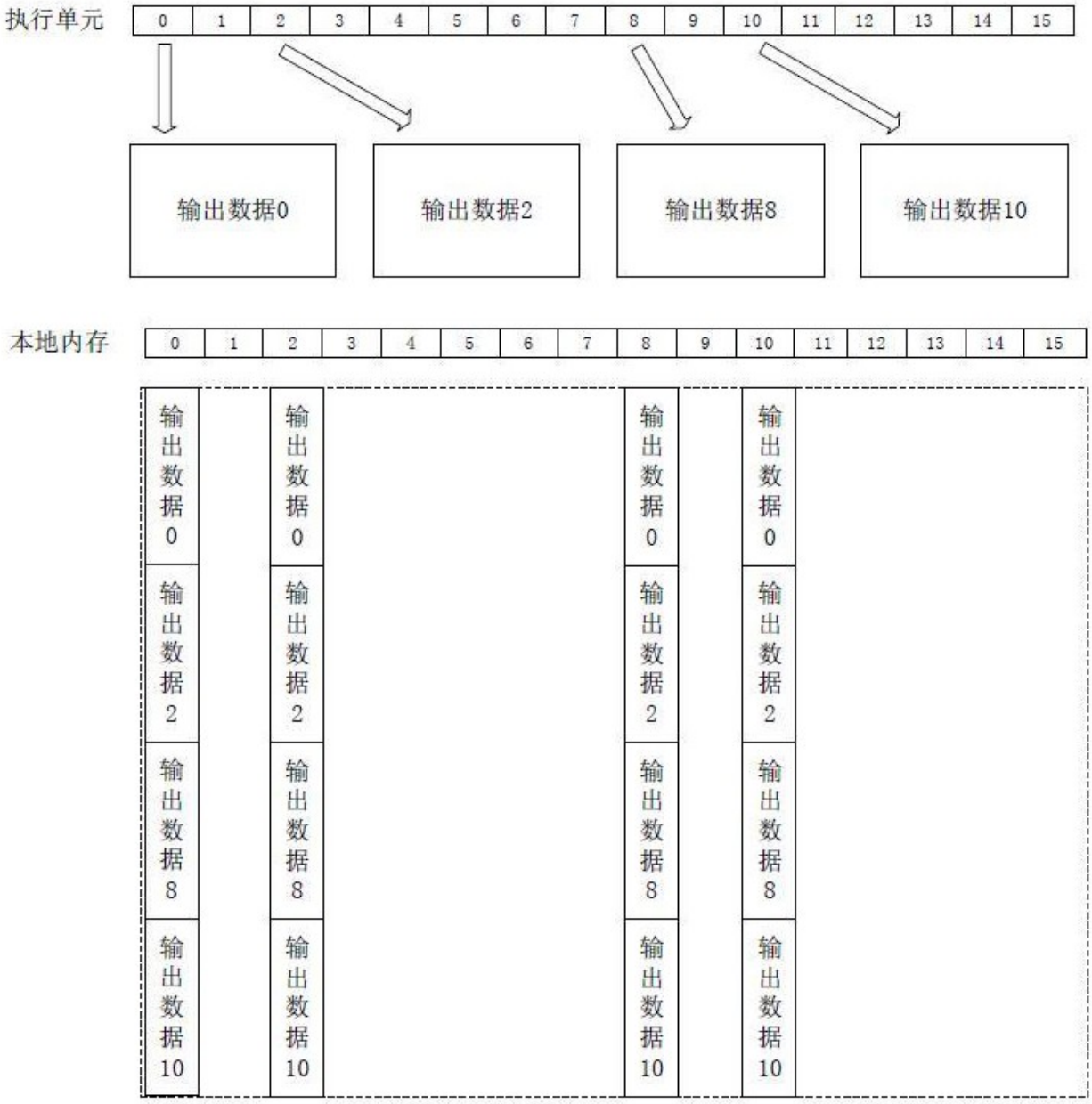

执行单元
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
输出数据0
输出数据2
输出数据8
输出数据10
本地内存
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
输出数据0
输出数据2
输出数据8
输出数据10
输出数据0
输出数据2
输出数据8
输出数据10
输出数据0
输出数据2
输出数据8
输出数据10
输出数据0
输出数据2
输出数据8
输出数据10

图 5

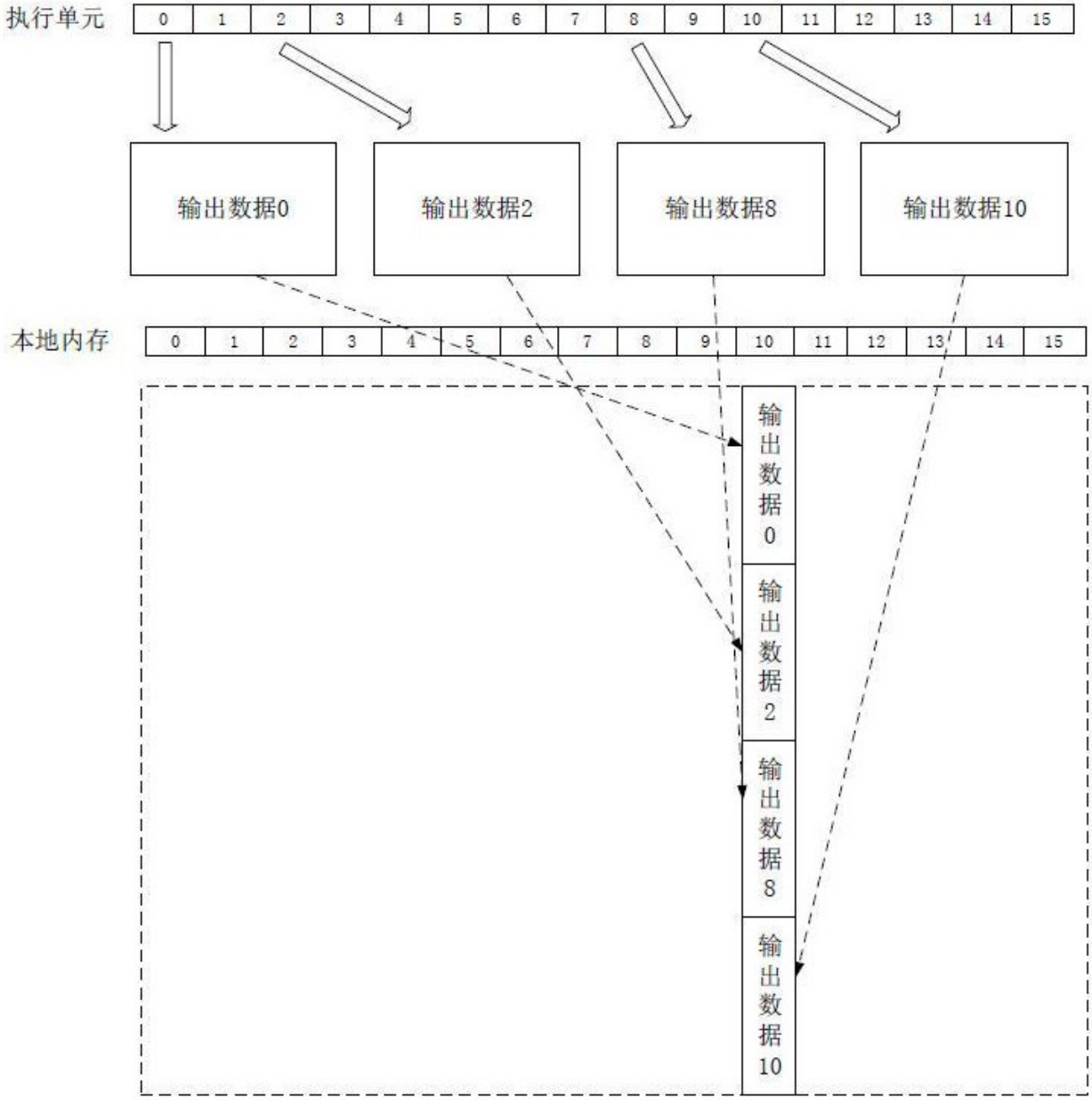

执行单元
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
输出数据0
输出数据2
输出数据8
输出数据10
本地内存
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
输出数据0
输出数据2
输出数据8
输出数据10

图 6

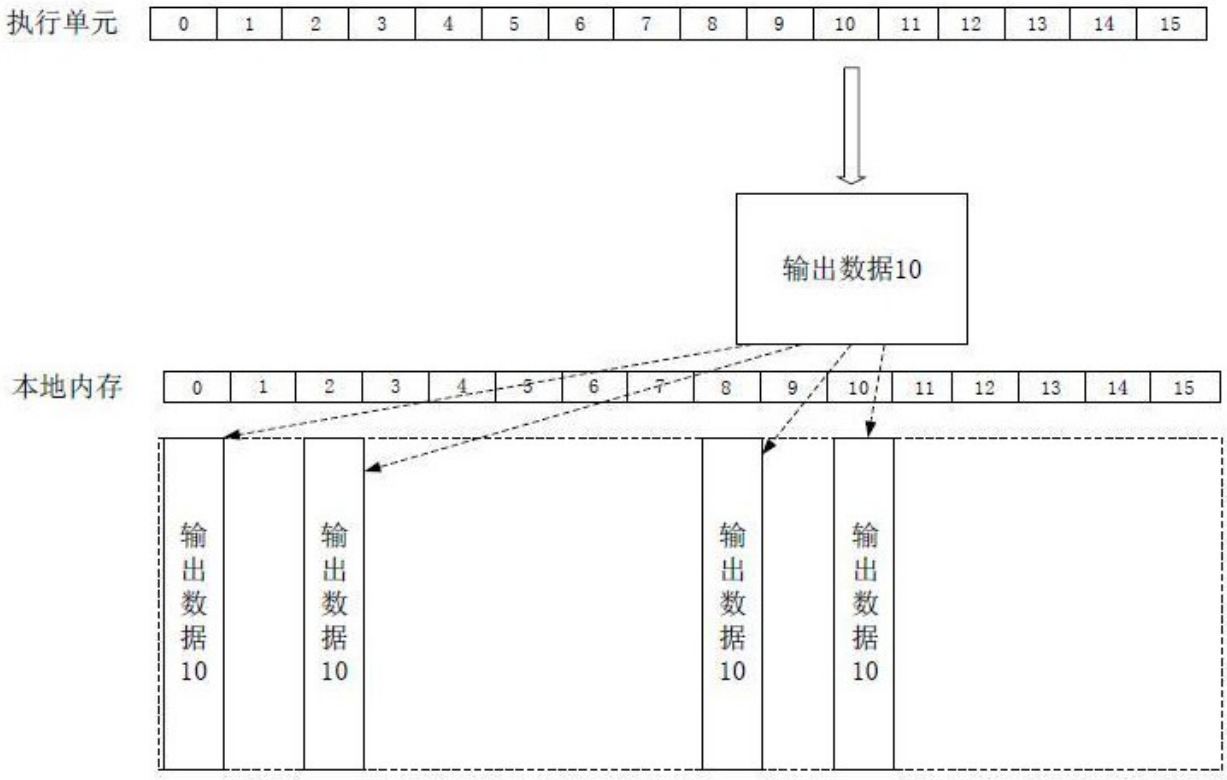

执行单元
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
输出数据10
本地内存
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
输出数据10
输出数据10
输出数据10
输出数据10

图 7

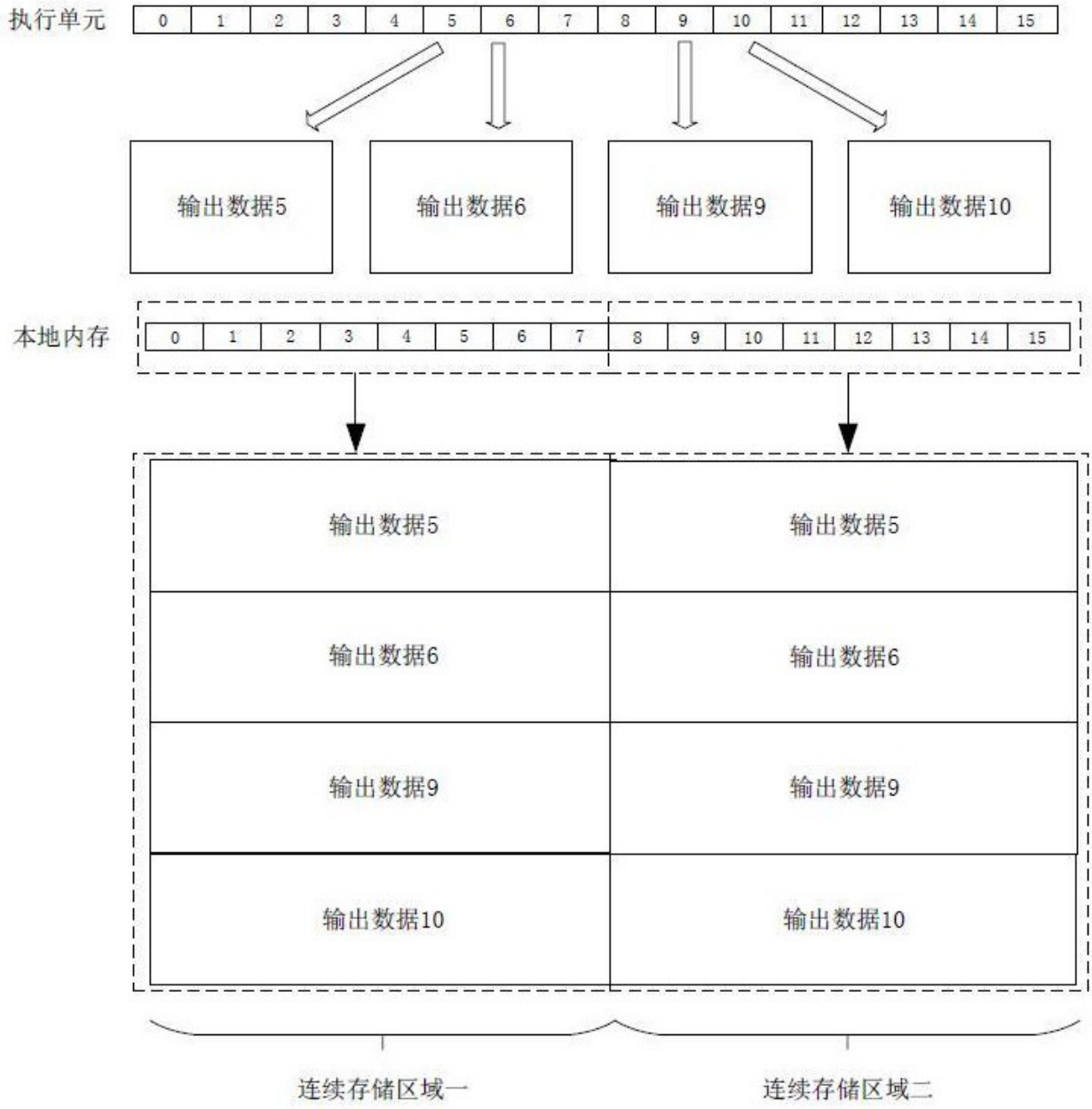

执行单元
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
输出数据5
输出数据6
输出数据9
输出数据10
本地内存
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
输出数据5
输出数据5
输出数据6
输出数据6
输出数据9
输出数据9
输出数据10
输出数据10
连续存储区域一
连续存储区域二

图 8