



(12) 发明专利

(10) 授权公告号 CN 113688064 B

(45) 授权公告日 2025.07.15

(21) 申请号 202010421447.1

G06N 3/0464 (2023.01)

(22) 申请日 2020.05.18

G06N 3/063 (2023.01)

(65) 同一申请的已公布的文献号

(56) 对比文件

申请公布号 CN 113688064 A

WO 2016191964 A1, 2016.12.08

(43) 申请公布日 2021.11.23

US 5659704 A, 1997.08.19

(73) 专利权人 中科寒武纪科技股份有限公司

审查员 张阿真

地址 100191 北京市海淀区知春路7号致真  
大厦D座16层1601房

(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

(74) 专利代理机构 北京维昊知识产权代理事务  
所(普通合伙) 11804

专利代理人 李波 孙新国

(51) Int.Cl.

G06F 12/02 (2006.01)

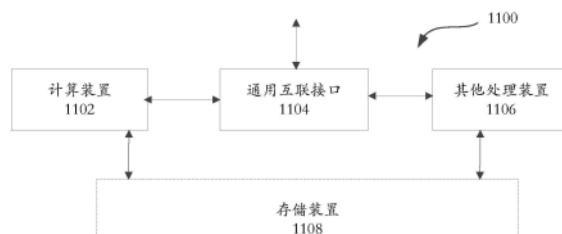
权利要求书3页 说明书18页 附图11页

(54) 发明名称

一种在存储器中为数据分配存储地址的方  
法和设备

(57) 摘要

本公开涉及一种在存储器中为数据分配存  
储地址的方法、设备和计算装置，其中计算装  
置可以包括在组合处理装置中，该组合处理装  
置还可以包括通用互联接口和其他处理装置。所述计  
算装置与其他处理装置进行交互，共同完成用户  
指定的计算操作。组合处理装置还可以包括存  
储装置，该存储装置分别与计算装置和其他处理装  
置连接，用于计算装置和其他处理装置的数据。  
本公开的技术方案能够提高存储器的存储空间  
利用率。



1.一种在存储器中为数据分配存储地址的方法,包括:

接收针对存储器中存储空间的空间分配请求,所述分配请求携带以第一函数表示的数据大小,所述第一函数中包括未知变量,所述未知变量仅在运行时才确定的;

根据接收到的所述空间分配请求,通过遍历树状结构来为数据分配目标存储空间,其中,所述树状结构包括多个节点,每个节点表示一个存储空间;以及

基于所分配的目标存储空间,通过宽度优先的策略遍历所述树状结构为所述数据分配存储地址。

2.根据权利要求1所述的方法,进一步包括:

形成多个存储块,每个存储块具有总存储空间标识和可用存储空间标识,其中总存储空间标识用于以总空间函数来指示所述存储块的总存储空间,所述可用存储空间标识用于以第二函数来指示所述存储块的可用存储空间;

将所述多个存储块作为节点形成树状结构,所述树状结构包括多级节点,其中上级节点的总存储空间为下级节点的总存储空间之和。

3.根据权利要求2所述的方法,其中,根据接收到的所述空间分配请求来为数据分配目标存储空间包括:

在所述树状结构的节点中检索可用存储空间;

响应于在所述树状结构的节点中搜索到其第二函数与所述第一函数具有公共函数的可用存储空间,将所述公共函数所指示的存储空间作为目标存储空间。

4.根据权利要求3所述的方法,其中,响应于在所述树状结构的节点中搜索到其第二函数与所述第一函数具有最大公共函数的可用存储空间,将所述公共函数所指示的存储空间作为目标存储空间。

5.根据权利要求3所述的方法,其中,重复在所述树状结构的节点中检索可用存储空间,直到检索不到可用存储空间为止。

6.根据权利要求3所述的方法,其中,根据接收到的所述空间分配请求来为数据分配目标存储空间包括:

响应于在所述树状结构的节点中未搜索到不小于所述数据大小的可用存储空间,创建新节点以使得可用存储空间能够容纳所述数据。

7.根据权利要求6所述的方法,其中,创建新节点以使得可用存储空间能够容纳所述数据包括:

将具有可用存储空间的节点调整为所述新节点的下级节点;

在所述新节点下创建新下级节点以使得所述新节点的下级节点的可用存储空间能够容纳所述数据;或者

创建新节点以使得可用存储空间能够容纳所述数据包括:

响应于所述树状结构中所有节点的可用存储空间为零,创建新节点以使得所述新节点的可用存储空间能够容纳所述数据;或者

创建新节点以使得可用存储空间能够容纳所述数据包括:

响应于在所述树状结构的节点中未搜索到不小于所述数据大小的当前可用存储空间,并且下一空间分配请求包括的数据大小不大于所述当前可用存储空间,则创建新节点以使得所述新节点的可用存储空间能够容纳所述数据。

8. 根据权利要求6所述的方法,其中,创建新节点以使得可用存储空间能够容纳所述数据包括:

响应于所述第一函数和第二函数均具有非公共部分,创建新节点;

将其第二函数与所述第一函数具有公共函数的节点调整为所述新节点的下级节点;以及,

在所述新节点下创建新下级节点以使得所述新节点的下级节点的可用存储空间能够容纳所述数据。

9. 根据权利要求3-8中任意一项所述的方法,其中,在所述树状结构的节点中检索可用存储空间包括:

在检索完一个上级节点以及该上级节点的所有下级节点之后,再检索与该上级节点同级的另外一个节点。

10. 根据权利要求2所述的方法,进一步包括:

调整所述节点的位置,以使得所述多个存储块的空间是连续的。

11. 根据权利要求2所述的方法,进一步包括:

固定节点在所述树状结构的中的排列顺序。

12. 根据权利要求2所述的方法,进一步包括:

响应于接收到空间释放请求,释放相应的节点;

更新被释放节点的可用存储空间标识符。

13. 根据权利要求2所述的方法,基于所分配的目标存储空间,为所述数据分配存储地址包括:

在为一个节点分配存储地址之后,依次为同级的其他节点分配地址,其中每个节点的地址涵盖该节点所有下级节点的地址。

14. 一种对存储空间进行管理的方法,包括:

形成多个存储块,每个存储块具有总存储空间标识和可用存储空间标识,其中总存储空间标识用于以总空间函数来指示所述存储块的总存储空间,所述可用存储空间标识用于以第二函数来指示所述存储块的可用存储空间,所述第二函数中包括未知变量,所述未知变量仅在运行时才确定的;

将所述多个存储块作为节点形成为树状结构,所述树状结构包括多级节点,其中上级节点的总存储空间为下级节点的总存储空间之和。

15. 一种在存储器中为数据分配存储空间的方法,包括:

接收针对存储器中存储空间的空间分配请求,所述分配请求携带以第一函数表示的数据大小,所述第一函数中包括未知变量,所述未知变量仅在运行时才确定的;

形成多个存储块,每个存储块具有总存储空间标识和可用存储空间标识,其中总存储空间标识用于以总空间函数来指示所述存储块的总存储空间,所述可用存储空间标识用于以第二函数来指示所述存储块的可用存储空间;

将所述多个存储块作为节点形成为树状结构,所述树状结构包括多级节点,其中上级节点的总存储空间为下级节点的总存储空间之和;

根据接收到的所述空间分配请求,根据所述树状结构来为数据分配目标存储空间。

16. 根据权利要求14或15所述的方法,进一步包括:

响应于接收到为数据分配目标存储空间的空间分配请求,在所述树状结构的节点中检索可用存储空间;

响应于在所述树状结构的节点中搜索到其第二函数与第一函数具有公共函数的可用存储空间,将所述公共函数所指示的存储空间作为目标存储空间。

17. 根据权利要求16所述的方法,其中,响应于在所述树状结构的节点中搜索到其第二函数与所述第一函数具有最大公共函数的可用存储空间,将所述公共函数所指示的存储空间作为目标存储空间。

18. 根据权利要求16所述的方法,其中,重复在所述树状结构的节点中检索可用存储空间,直到检索不到可用存储空间为止。

19. 根据权利要求16所述的方法,其中,响应于在所述树状结构的节点中未搜索到不小于所述数据大小的可用存储空间,创建新节点以使得可用存储空间能够容纳所述数据。

20. 根据权利要求19所述的方法,其中,创建新节点以使得可用存储空间能够容纳所述数据包括:

将具有可用存储空间的节点调整为所述新节点的下级节点;

在所述新节点下创建新下级节点以使得所述新节点的下级节点的可用存储空间能够容纳所述数据;或者

创建新节点以使得可用存储空间能够容纳所述数据包括:

响应于所述树状结构中所有节点的可用存储空间为零,创建新节点以使得所述新节点的可用存储空间能够容纳所述数据;或者

创建新节点以使得可用存储空间能够容纳所述数据包括:

响应于在所述树状结构的节点中未搜索到不小于所述数据大小的当前可用存储空间,并且下一空间分配请求包括的数据大小不大于所述当前可用存储空间,则创建新节点以使得所述新节点的可用存储空间能够容纳所述数据。

21. 根据权利要求16所述的方法,其中,在所述树状结构的节点中检索可用存储空间包括:

在检索完一个上级节点以及该上级节点的所有下级节点之后,再检索与该上级节点同级的另外一个节点。

22. 根据权利要求14或15所述的方法,进一步包括:

调整所述节点的位置,以使得所述多个存储块的空间是连续的。

23. 根据权利要求14或15所述的方法,进一步包括:

固定节点在所述树状结构中的排列顺序。

24. 一种电子设备,包括:

一个或多个处理器;以及

存储器,所述存储器中存储有计算机可执行指令,当所述计算机可执行指令由所述一个或多个处理器运行时,使得所述电子设备执行如权利要求1-23中任意一项所述的方法。

25. 一种计算机可读存储介质,包括计算机可执行指令,当所述计算机可执行指令由一个或多个处理器运行时,执行如权利要求1-23中任意一项所述的方法。

## 一种在存储器中为数据分配存储地址的方法和设备

### 技术领域

[0001] 本发明涉及计算机领域,更具体地,涉及存储空间的管理。

### 背景技术

[0002] 深层神经网络(DNN)已广泛用于各种应用,例如计算机视觉,语音识别,机器翻译和机器人技术,因为与传统的机器学习方法相比,其准确性有较大提高。但是,DNN的性能优势是以极高的计算和内存复杂性为代价的,底层硬件架构面临的挑战。为了提高DNN处理的效率,已经提出了各种专用加速器,以提供能够比通用架构(例如CPU和GPU)更好的性能和效能。

[0003] 为了更好地利用存储空间,存在多种对存储器的空间进行管理的方案,如最佳匹配法(best fit,BF)和图形着色(graph coloring,GC)等。

[0004] 但是,现有技术方案中均是以立即数的形式来分配存储空间的,对于可变大小的数据尚无合适的方案。而且,现有技术中对空间的利用也并不充分。

### 发明内容

[0005] 本公开的一个目的是为未知大小的、以函数方式表示的数据来分配合适的存储空间。

[0006] 根据本公开的第一方面,提供一种在存储器中为数据分配存储地址的方法,包括:接收针对存储器中存储空间的空间分配请求,所述分配请求携带以第一函数表示的数据大小;根据接收到的所述空间分配请求来为数据分配目标存储空间;以及基于所分配的目标存储空间,为所述数据分配存储地址。

[0007] 根据本公开的第二方面,提供一种对存储空间进行管理的方法,包括:形成多个存储块,每个存储块具有总存储空间标识和可用存储空间标识,其中总存储空间标识用于以总空间函数来指示所述存储块的总存储空间,所述可用存储空间标识用于以第二函数来指示所述存储块的可用存储空间;将所述多个存储块作为节点形成为树状结构,所述树状结构包括多级节点,其中上级节点的总存储空间为下级节点的总存储空间之和。

[0008] 根据本公开的第三方面,提供一种在存储器中为数据分配存储空间的方法,包括:接收针对存储器中存储空间的空间分配请求,所述分配请求携带以第一函数表示的数据大小;形成多个存储块,每个存储块具有总存储空间标识和可用存储空间标识,其中总存储空间标识用于以总空间函数来指示所述存储块的总存储空间,所述可用存储空间标识用于以第二函数来指示所述存储块的可用存储空间;将所述多个存储块作为节点形成为树状结构,所述树状结构包括多级节点,其中上级节点的总存储空间为下级节点的总存储空间之和;根据接收到的所述空间分配请求,根据所述树状结构来为数据分配目标存储空间。

[0009] 根据本公开的第四方面,提供一种电子设备,包括:一个或多个处理器;以及存储器,所述存储器中存储有计算机可执行指令,当所述计算机可执行指令由所述一个或多个处理器运行时,使得所述电子设备执行如上所述的方法。

[0010] 根据本公开的第五方面，提供一种计算机可读存储介质，包括计算机可执行指令，当所述计算机可执行指令由一个或多个处理器运行时，执行如上所述的方法。

## 附图说明

[0011] 通过参考附图阅读下文的详细描述，本公开示例性实施方式的上述以及其他目的、特征和优点将变得易于理解。在附图中，以示例性而非限制性的方式示出了本公开的若干实施方式，并且相同或对应的标号表示相同或对应的部分其中：

- [0012] 图1a示出了一般的神经网络运算示意图；
- [0013] 图1b示出了根据图1a的运算过程分配存储空间的一种示例；
- [0014] 图1c示出了根据图1a的运算过程分配存储空间的另外一种示例；
- [0015] 图2示出了根据本公开一个实施方式的在存储器中为数据分配存储地址的方法流程图；
- [0016] 图3示出了建立该树状结构的一种方法的流程图；
- [0017] 图4a示出了根据本公开一个实施方式的树状结构示意图；
- [0018] 图4b示出了根据本公开一个实施方式的树状结构示意图；
- [0019] 图4c示出了根据本公开一个实施方式的树状结构示意图；
- [0020] 图4d示出了根据本公开的另外一个实施方式的为数据分配空间的示意图；
- [0021] 图5示出了根据本公开的优选实施方式一个实施方式的为数据分配目标存储空间的示意图；
- [0022] 图6a示出了根据本公开的一个实施方式的创建新节点的方法流程图；
- [0023] 图6b示出了根据图6a所示的方法流程图创建的新节点的示意图；
- [0024] 图6c示出了根据图6a所示的方法流程图创建的新节点的示意图；
- [0025] 图7a示出了根据本公开的一个实施方式的建立新节点的流程图；
- [0026] 图7b示出了根据本公开的一个实施方式的建立新节点的示意图；
- [0027] 图8示出了根据本公开的一个实施方式的为存储空间分配地址的示意图；
- [0028] 图9示出了根据本公开的一个实施方式的对存储空间进行管理的方法；
- [0029] 图10示出了根据本公开的一个实施方式的在存储器中为数据分配存储空间的方法；
- [0030] 图11示出了一种组合处理装置的示意性框图；以及
- [0031] 图12示出了一种板卡的示意性框图。

## 具体实施方式

[0032] 下面将结合本披露实施例中的附图，对本披露实施例中的技术方案进行清楚、完整地描述，显然，所描述的实施例是本披露一部分实施例，而不是全部的实施例。基于本披露中的实施例，本领域技术人员在没有做出创造性劳动前提下所获得的所有其他实施例，都属于本披露保护的范围。

[0033] 应当理解，本披露的权利要求、说明书及附图中的术语“第一”、“第二”、“第三”和“第四”等是用于区别不同对象，而不是用于描述特定顺序。本披露的说明书和权利要求书中使用的术语“包括”和“包含”指示所描述特征、整体、步骤、操作、元素和/或组件的存在，

但并不排除一个或多个其它特征、整体、步骤、操作、元素、组件和/或其集合的存在或添加。

[0034] 还应当理解,在此本披露说明书中所使用的术语仅仅是出于描述特定实施例的目的,而并不意在限定本披露。如在本披露说明书和权利要求书中所使用的那样,除非上下文清楚地指明其它情况,否则单数形式的“一”、“一个”及“该”意在包括复数形式。还应当进一步理解,在本披露说明书和权利要求书中使用的术语“和/或”是指相关联列出的项中的一个或多个的任何组合以及所有可能组合,并且包括这些组合。

[0035] 如在本说明书和权利要求书中所使用的那样,术语“如果”可以依据上下文被解释为“当...时”或“一旦”或“响应于确定”或“响应于检测到”。类似地,短语“如果确定”或“如果检测到[所描述条件或事件]”可以依据上下文被解释为意指“一旦确定”或“响应于确定”或“一旦检测到[所描述条件或事件]”或“响应于检测到[所描述条件或事件]”。

[0036] 图1a-图1c示出了多种不同的内存分配方法的示意图,其中图1a示出了一般的神经网络运算示意图。

[0037] 如图1a所示,该卷积神经网络包括了数据D1-D4,符号Conv表示卷积运算,由此D2=Conv(D1),D3=Conv(D2)以及D4=Conv(D3),其中,例如D1为16比特,D2为8比特,D3为24比特,D4为32比特。在不断地进行卷积运算过程中,需要为参与运算的数据分配存储空间,并且将历史数据占用的空间释放。

[0038] 图1b示出了根据图1a的运算过程分配存储空间的一种示例,该示例称为最佳匹配法(best fit,BF)。

[0039] 在图1b中,首先为数据D1和D2分配地址;当进行第二个卷积运算那时,可以释放D1所占据的空间,并且为数据D3分配存储空间;类似地,当进行第三个卷积运算时,可以释放D2的空间,并且进一步为D4分配存储空间。在1b所示的方案中,将会产生大量的空间碎片,从而降低了存储空间的利用率。

[0040] 图1c示出了根据图1a的运算过程分配存储空间的另外一种示例,该示例称为图形着色(graph coloring,GC)。

[0041] 如图1c所示,存储空间被划分为多个类,Regclass32表示容量为32比特的空间,Regclass24表示容量为24比特的空间,Regclass16表示容量为16比特的空间,而Regclass8表示容量为8比特的空间。在此情况下,数据D1-D4将根据大小的不同被划分到不同的分类中,例如数据D1被存储在Regclass16中的R<sub>16,0</sub>空间,数据D2被存储在Regclass8中的R<sub>8,4</sub>空间,数据D3被存储在Regclass24中的R<sub>24,0</sub>空间,而数据D4被存储在Regclass32中的R<sub>32,1</sub>空间。在这种分配方式下,仍然会产生大量的碎片,特别是当某个数据的大小小于某个类而大于另外一个类时,则存储空间的碎片化更加明显。例如,如果某数据的大小为20比特,则其应当存储在Regclass24,由此造成4比特的空间浪费。

[0042] 图2示出了根据本公开的一个实施方式的在存储器中为数据分配存储地址的方法,包括:在操作S210,接收针对存储器中存储空间的空间分配请求,所述分配请求携带以第一函数表示的数据大小;在操作S220,根据接收到的所述空间分配请求来为数据分配目标存储空间;以及,在操作 S230,基于所分配的目标存储空间,为所述数据分配存储地址。

[0043] 需要理解的是,在上文中,空间分配请求中除了数据大小之外,还可以携带其他信息,例如资源的生存周期,该生存周期表示资源什么时候申请,什么时候释放,以及申请及释放的顺序等等。不同的顺序或导致不同的结果。在本公开中,数据的并不是立即数,而是

以一个函数来表示的,函数中包含有未知变量,而这些未知变量只有在运行时(runtime)才是确定的,而在为这些数据分配存储空间时并不是确定的数值。

[0044] 下面以一个四维数据为例,其大小可以表示为 $S=N*H*W*C$ ,其中,参数N,H,W和C均为正自然数,并且这四个参数中至少有一个是可变的。为了更方便理解,若假设初始网络输入大小为 $1*H*W*3$ ,其中,H和W是未知参数。经过神经网络的多个层之后,到了第X层,该层输入大小变为 $1*f1(H)*f2(W)*3$ 。此时的f是根据从第一层到第 X层的算子的参数决定的。此时有 $f1(H)>0,f2(W)>0$ 。由于在神经网络中,f1和f2均为最高一次的多项式,因此我们可以根据该不等式,得到参数H和W最小大小的条件,例如 $H>x1,W>x2$ 。

[0045] 对于整个网络运算下来,得到整个神经网络对HW的约束。可以设 $H = h+x1, W=w+x2$ ,将该式代入,原网络中每一个数据块表达式就只有+号了,其中x1和x2是神经网络约束的最小值,由于网络结构需要保证计算到最后一层大小不能为0,因此对于整个网络运算而言,需要有对H和 W的约束。由此,数据的大小可以例如表示为 $P1hw+P2h+P3w+P4$ ,其中P1,P2,P3和P4均为非负整数。

[0046] 下面是一组示例性空间分配指令和空间释放指令,以便于对本公开的技术方案有更清楚的理解,其中操作1至操作8按时间顺序执行。

[0047] 操作1:Alloc A 10hw+5h+5w+20

[0048] 操作2:alloc A1 9hw+4h+6w+5

[0049] 操作3:alloc A2 hw+h+10w+8

[0050] 操作4:Dealloc A

[0051] 操作5:alloc B 8hw+10h+10w+100

[0052] 操作6:alloc C 4hw+20h+20w+50

[0053] 操作7:dealloc B

[0054] 操作8:alloc D 12hw+5h+5w+10

[0055] 指令“alloc”表示分配存储空间,而指令“dealloc”表示释放存储空间。例如,操作1“Alloc A 10hw+5h+5w+20”表示为数据A分配大小为 $10hw+5h+5w+20$ 的空间,操作5“alloc B 8hw+10h+10w+100”表示为数据B分配大小为 $8hw+10h+10w+100$ 等,而操作4“Dealloc A”表示释放数据A所占用的空间,操作7“dealloc B”表示释放数据B 所占用的空间。

[0056] 接下来,在接收到该空间分配请求之后,可以从可用存储空间中选择适当的一部分或者全部,作为目标存储空间来分配给上述的数据。需要解释的是,上面的术语“目标存储空间”是指实际分配给数据的空间。这样的描述仅用于在上下文中区分所指代的不同空间以便于理解,而不对本公开的技术方案造成任何的限制。需要理解的是,上面的“可用存储空间”和“目标存储空间”是一种虚拟的存储空间,可以将这些虚拟的存储空间对应到相应的数据,然后,将这些虚拟存储空间映射到实际的物理空间和地址。但上下文中为了表示这些虚拟存储空间的含义,以不同的名称来进行表述。

[0057] 确定了存储空间之后,则将相应的存储地址分配给数据。存储地址可以包括起始地址和偏移量,偏移量可以是由数据的大小来决定。

[0058] 其中,将相应的存储地址分配给数据的过程可以是将虚拟的存储空间映射到实际的物理空间上的存储地址的过程。在上文中,为数据分配目标存储空间的时候,将查找现有可用存储空间,以分配适当的那部分存储空间给该数据。在分配存储空间的过程中,暂时并

不分配具体的存储地址。这种空间分配和地址分配相分离的方法,显著地区别于现有技术中一步就将存储空间和存储地址分配给数据的做法。

[0059] 根据本公开的一个实施方式,可以建立一种树状结构,该树状结构可以包括多个节点,每个节点表示一个存储空间,可以通过遍历该存储空间来找到适合的节点来存储数据。这里所述的“适合的节点”,可以是一个节点,也可以是多个节点的组合。

[0060] 需要理解的是,建立存储块之间的树状关系可以在上述的操作S210 之前实现,即可以预先建立下文中将要详细描述的树状结构,也可以在操作S210之后,在操作S220之前实现,即通过空间分配请求以及空间释放的操作实时地建立这种树状结构。

[0061] 图3示出了建立该树状结构的一种方法的流程图。

[0062] 如图3所示,本公开的方法进一步包括:在操作S310,形成多个存储块,每个存储块具有总存储空间标识和可用存储空间标识,其中总存储空间标识用于以总空间函数来指示所述存储块的总存储空间,所述可用存储空间标识用于以第二函数来指示所述存储块的可用存储空间;在操作 S320,将所述多个存储块作为节点形成为树状结构,所述树状结构包括多级节点,其中上级节点的总存储空间为下级节点的总存储空间之和。

[0063] 上面的操作1-操作8以及图4a-图4c来描述树状结构的建立过程。

[0064] 如图4a所示,首先进行初始化,以节点0作为初始节点,其中每个节点表示一个存储块,其总存储空间为0,可用存储空间为0,为方便描述,其存储属性可以表示为 $0(0)$ 。

[0065] 接下来,当接收到操作1的指令Alloc A 10hw+5h+5w+20时,在节点0下建立节点1,用于存储数据A。因此,该节点1的总存储空间为 10hw+5h+5w+20,可用存储空间为0,其存储属性可以表示为 $10hw+5h+5w+20(0)$ 。

[0066] 当接收到操作2的指令alloc A1 9hw+4h+6w+5时,在节点0下建立节点2,用于存储数据A1。因此,该节点2的总存储空间为 $9hw+4h+6w+5$ ,可用存储空间为0,其存储属性可以表示为 $9hw+4h+6w+5(0)$ 。

[0067] 当接收到操作3的指令alloc A2 hw+h+10w+8时,在节点0下建立节点3,用于存储数据A2。因此,该节点3的总存储空间为 $hw+h+10w+8$ ,可用存储空间为0,其存储属性可以表示为 $hw+h+10w+8(0)$ 。

[0068] 此时,初始节点0的总存储空间为 $20hw+10h+21w+33$ ,其可用存储空间为0,其存储属性可以表示为 $20hw+10h+21w+33(0)$ 。

[0069] 可以理解的是,还可以释放存储空间,根据本公开的一个实施方式,响应于接收到空间释放请求,释放相应的节点;更新被释放节点的可用存储空间标识符。

[0070] 以图4b为例来进行描述。当接收到操作4的指令dealloc A时,将节点1的空间释放,从而节点1的可用存储空间为 $10hw+5h+5w+20$ ,其存储属性可以表示为 $10hw+5h+5w+20(10hw+5h+5w+20)$ ,如图4b 所示。此时,初始节点0的总存储空间为 $20hw+10h+21w+33$ ,其可用存储空间标识仍然为0,其存储属性可以表示为 $20hw+10h+21w+33(0)$ 。

[0071] 在本公开中,所释放的空间可以被再次使用,以达到空间被充分利用的目的。由此,根据本公开的一个实施方式,据接收到的所述空间分配请求来为数据分配目标存储空间包括:在所述树状结构的节点中检索可用存储空间;响应于在所述树状结构的节点中搜索到其第二函数与所述第一函数具有公共函数的可用存储空间,将所述公共函数所指示的存储空间作为目标存储空间。这里的“在所述树状结构的节点中搜索到其第二函数”可以是

单个节点对应的第二函数,也可以是一个以上的连续节点组合形成的“第二函数”。

[0072] 需要理解的是,对于以函数表示的数据大小,两个函数具有公共部分包括如下几种情况:

[0073] 第一种情况是指第一函数与第二函数具有部分重叠和交叉部分,但彼此并不完全覆盖对方;第二种情况是第一函数完全覆盖并且大于第二函数,即空间分配请求包含的数据大小完全大于所搜索到的可用存储空间的大小;第三种情况是第二函数完全覆盖且大于第一函数,其空间分配请求包含的数据大小完全小于所搜索到的可用存储空间的大小;第四种是第一函数和第二函数完全相等。

[0074] 以图4c为例来说明根据普遍性和复杂性的第一种情况。例如,当接收到操作5的指令alloc B 8hw+10h+10w+100时,表示要为数据B请求分配大小为8hw+10h+10w+100的存储空间。此时仅有节点1中具有可用存储空间10hw+5h+5w+20,该可用存储空间的函数10hw+5h+5w+20与要分配的空间的函数8hw+10h+10w+100具有公共函数 8hw+5h+5w+20,而在将8hw+5h+5w+20大小的空间分配给数据B的公共部分之后,节点1的可用存储空间为2hw,而数据B仍然需要5h+5w+80 的存储空间。

[0075] 在此情况下,对于数据B而言,其公共部分可以按照8hw+5h+5w+20 的方式进行存储,另外一部分,即空间分配请求中的非公共部分,5h+5w+80 可以另外创建节点(例如图4c 所示的节点5)来存储。需要理解的是,可以将数据B的公共部分直接存储在节点1中大小为8hw+5h+5w+20的存储空间中,这种方式无需创建新的节点,但由于节点1中原先分配给数据A 的存储空间被分配给了数据B的公共部分,因此原先节点1与数据A之间的对应关系将被擦除,这不利于后续存储关系的保存。

[0076] 根据本公开的一个实施方式,可以在节点1的下级建立新节点4,以使得新节点的总存储空间等于上述的公共函数。以图4c为例,新节点4 的存储属性可以表示为8hw+5h+5w+20(0)。此时,节点1的可用存储空间为2hw,从而节点1的存储属性可以表示为10hw+5h+5w+20(2hw)。此外,节点0的总存储空间标识符变为20hw+15h+26W+113。如图4c所示。节点1的可用存储空间2hw可以继续分配给其他数据。

[0077] 由此,根据本公开的一个实施方式,在逻辑层面上,同一个节点只能分配一次,以保证同一个节点对应一个数据,这样可保证整个分配过程都被完整地记录下来,也便于后续进一步拆分子节点。例如节点1由于之前已经分配给数据A,即使数据A占用的空间被释放,当有新的数据(例如数据B)需要占用数据A原先占用的空间时,在逻辑上仍然要分出新的节点(例如节点4)。当然,在物理空间上,新创建的节点4仍然在节点1 的物理空间的范围之内。

[0078] 此外,需要解释的是,新创建的节点5与节点4紧邻,以尽量使得数据B能够存储在连续的空间中。根据本公开的另一个实施方式,节点5可以处于节点1的左边,由此节点5与节点4紧邻并且具有连续的空间。需要理解的是,图4c仅仅是一种示例,新节点的创建会根据实际的需要来调整位置,以确保同一数据能够存储在连续的空间中。这将在后文中进行更加详细的描述。此外,同一数据能够存储在连续的空间中仅仅是一种优选的情形,而不是强制的要求,例如,同一数据存放也可以不连续。

[0079] 接下来,当执行操作6 “alloc C 4hw+20h+20w+50”时,可以首先复用节点1中的可用存储空间2hw,然后再申请一个新的空间2hw+20h+ 20w+50。类似地,可以创建新的节点,

该新的节点具有总存储空间  $2hw+20h+20w+50$ 。

[0080] 接下来,当执行操作7“dealloc B”时,释放数据B所占用的空间,从而可用存储空间为 $8hw+10h+10w+100$ 。

[0081] 接下来,当执行操作8“alloc D 12hw+5h+5w+10”时,可以复用的空间为 $8hw+5h+5w+10$ ,还需要额外申请 $4hw$ 的空间。

[0082] 图4d示出了根据本公开的另外一个实施方式的为数据分配空间的示意图。

[0083] 在图4d中,节点0的总存储空间为 $14hw+14h+22w+260$ ,其具有四个下级节点,即节点1( $2hw+3h+2w+80$ )、节点2( $3hw+h+5w$ )、节点3( $4hw+4h+7w+80$ (0))以及节点4( $5hw+6h+8w+100$ )。此时收到操作指令alloc B $14hw+4h+7w+80$ ,在上述四个节点中,节点1和节点2的可用空间之和为 $5hw+4h+7w+80$ ,大于数据B1的大小并足以容纳数据B1,并且与数据B1的数据大小的函数之差为 $hw$ ;节点3的可用存储空间为0,无法容纳数据B1;节点4的可用存储空间为 $5hw+6h+8w+100$ ,大于数据B1的大小并足以容纳数据B1。在此情况下,可以有多种情形以存储该数据B1。

[0084] 第一种方案为,从利用节点4为该数据B1分配 $4hw+4h+7w+80$ 的存储空间,从而在分配完之后,节点4会出现 $hw+h+w+20$ 的可用存储空间。

[0085] 第二种方案为,占用节点1中的可用存储空间 $2hw+3h+2w+80$ ,然后再从节点2或节点4中分配 $2hw+h+5w$ 的存储空间。

[0086] 第三种方案为,占用节点2中的可用存储空间 $3hw+h+5w$ ,然后再从节点1或节点4中分配 $hw+3h+2w+80$ 的存储空间。

[0087] 第四种方案为,可以分别从节点1、节点2和节点4中共分配大小为 $4hw+4h+7w+80$ 的存储空间,例如,可以分别从节点1中分配大小为 $2hw+2h+2w+40$ 的存储空间,从节点2中分配大小为 $2hw+h+5w$ 的存储空间,以及从节点4中分配大小为 $h+40$ 的存储空间。

[0088] 采用第一种方案的有益效果在于,能够更容易使得新分配的空间在地址上保持连续,减少数据存取过程中的地址跳变,从而提升数据存取的效率。

[0089] 采用第二和第三种方案的有益效果在于,能够尽量减少碎片空间的存在。此外,利用节点1和节点2来向数据分配存储空间的方式也有利于使得数据能够存储在连续的空间中,从而减少数据存取过程中的地址跳变。

[0090] 根据本公开的一个实施方式,响应于在所述树状结构的节点中未搜索到其可用存储空间能覆盖所述数据的单个节点,创建新节点以使得所述单个节点与新节点的可用存储空间之和不小于所述数据。

[0091] 仍然以图4d为例来进行说明。

[0092] 在第五种方案中,节点1的可用存储空间不能完全覆盖数据B1,可以先分配节点1中的可用存储空间 $2hw+3h+2w+80$ ,然后再申请新的存储空间 $2hw+h+5w$ ,从而总的存储空间为 $4hw+4h+7w+80$ ,足以容纳数据B1。

[0093] 在第六种方案中,节点2的可用存储空间不能完全覆盖数据B1,可先分配节点2中的可用存储空间 $3hw+h+5w$ ,然后再申请新的存储空间 $hw+3h+2w+80$ ,从而总的存储空间为 $4hw+4h+7w+80$ ,足以容纳数据B1。

[0094] 优选地,该新节点可以与上述的单个节点形成连续的空间。

[0095] 相较于第二种和第三种方案,第五种方案和第六种方案的有益效果在于,如果节

点1和节点2中的空间无法连续,那么数据存取过程中的空间跳变容易引起效率低下,而第五种和第六种方案则可以通过直接创建新的节点来形成连续的空间。

[0096] 图5示出了根据本公开的优选实施方式一个实施方式的为数据分配目标存储空间的示意图。

[0097] 根据本公开的一个实施方式,响应于在所述树状结构的节点中搜索到其第二函数与所述第一函数具有最大公共函数的可用存储空间,将所述公共函数所指示的存储空间作为目标存储空间。

[0098] 如图5所示,节点0具有六个下级节点,即节点1( $2hw+3h+2w+80$  ( $2hw+3h+2w+80$ ))、节点2( $3hw+h+5w$  ( $3hw+h+5w$ ))、节点3 ( $4hw+4h+7w+80$  ( $4hw+4h+7w+80$ )))、节点4( $4hw+5h+7W+80$  ( $4hw+5h+7W+80$ )))、节点5( $4hw+4h+8w+80$  ( $4hw+4h+8w+80$ )))以及节点6( $4hw+4h+7w+90$  ( $4hw+4h+7w+90$ )))。此时收到操作指令alloc B1  $4hw+4h+7w+80$ ,在上述六个节点中,节点1和节点2的可用空间之和为 $5hw+4h+7w+80$ ,大于数据B1的大小并足以容纳数据B1,并且与数据B1的数据大小的函数之差为 $hw$ ;节点3的可用存储空间为  $4hw+4h+7w+80$ ,刚好容纳数据B1;节点4的可用存储空间为  $4hw+5h+7W+80$ ,大于数据B1的大小并足以容纳数据B1,并且与数据B1 的数据大小的函数之差为 $h$ ;节点5的可用存储空间为 $4hw+4h+8w+80$ ,大于数据B1的大小并足以容纳数据B1,并且与数据B1的数据大小的函数之差为 $w$ ;节点6的可用存储空间为  $4hw+4h+7w+90$ ,大于数据B1的大小并足以容纳数据B1,并且与数据B1的数据大小的函数之差为10;根据本公开的一个实施方式,在节点1和节点2的组合、节点3、节点4、节点5以及节点6都足以容纳数据B1的情况下,可以优先选择节点3来容纳数据B1,因为其可用存储空间与数据B1的大小之差最小。上述优选实施方式有利于减少存储空间的碎片,提高存储空间利用效率。

[0099] 这里,我们将可用存储空间与待分配空间的数据的大小之差称为代价函数。对于节点1和2的组合,节点4,节点5和节点6,他们与数据B1 的大小之差(即代价函数)分别为 $hw$ , $h$ , $w$ 和10。在这种大小之差差别不大的情况下,判断第一函数和第二函数之间是否具有最大公共函数可以通过如下几种方式:第一,可以随机的选择其中一个节点(例如,节点1和2的组合,节点4,节点5或节点6),作为分配给该数据的节点;第二,可以将其第二函数与第一函数之差的某个项最小的节点,作为分配给该数据的节点,例如,在上面四个项目中,如果设大小之差中 $h$ 项和 $w$ 项最小的第二函数作为分配给该数据的节点,那么节点6可以被选择来分配给该数据。这种设置方式灵活多变,本领域技术人员可以根据实际需要来进行设置。

[0100] 根据本公开的一个实施方式,其中,重复在所述树状结构的节点中检索可用存储空间,直到检索不到可用存储空间为止。

[0101] 在该实施方式中,当某个节点的可用存储空间不足以满足空间分配请求中的数据大小时,则可以遍历该树状结构中的所有节点,以充分利用所有的空间碎片。例如,在图4c 中,节点1中尚有可用存储空间 $2hw$ ,因此,如果后续的空间分配请求中的数据大小包含了 $hw$ 这样的项,则仍然可以对该空间 $2hw$ 进行复用。当然,如果检索不到任何与第一函数具有公共函数的、以第二函数表示的可用存储空间,则可以构建新的节点。

[0102] 上面介绍了创建树状结构的情形,还存在当前所有节点的可用存储节点之和都不足以存储数据的情形,即第一函数完全覆盖并且大于第二函数,即空间分配请求包含的数

据大小完全大于所搜索到的可用存储空间的大小。根据本公开的一个实施方式,根据接收到的所述空间分配请求来为数据分配目标存储空间包括:响应于在所述树状结构的节点中未搜索到不小于所述数据大小的可用存储空间,创建新节点以使得可用存储空间能够容纳所述数据。

[0103] 下面将详细介绍多种创建新节点的实施方式。

[0104] 图6a示出了根据本公开的一个实施方式的创建新节点的方法流程图;图6b和图6c示出了根据图6a所示的方法流程图创建的新节点的示意图。

[0105] 如图6a所示,创建新节点以使得可用存储空间能够容纳所述数据可以包括:在S610,将具有可用存储空间的节点调整为所述新节点的下级节点;以及,在操作S620,在所述新节点下创建新下级节点以使得所述新节点的下级节点的可用存储空间能够容纳所述数据。其中,具有可用存储空间的节点不足以存储数据。

[0106] 假设当前的树状结构如图6b所示,其中节点0的总存储空间为 $20hw+15h+26w+113$ ,节点1、节点2和节点3是节点0的下级节点,其中节点1的总存储空间为 $10hw+10h+10w+100$ ,可用存储空间为0;节点2的总存储空间为 $9hw+4h+6w+5$ ,可用存储空间为 $9hw+4h+6w+5$ ;节点3的总存储空间为 $hw+h+10w+8$ ,可用存储空间为 $h+2w+5$ 。节点4是节点3的下级节点,其总存储空间为 $hw+8w+3$ ,可用存储空间为0。因此,此时的总可用存储空间为节点2的可用存储空间和节点3的可用存储空间之和: $9hw+5h+8w+10$ 。

[0107] 假设此时接收到一个空间分配请求,alloc D5 15hw+9h+10w+40,总可用存储空间 $9hw+5h+8w+10$ 小于该空间分配请求,尚需要 $6hw+4h+2w+30$ 的空间。在此情况下,需要创建新的节点。

[0108] 由此,如图6c所示,创建了与节点1等级别的节点8,并将原节点2和节点3移动到新节点8的下级作为新节点8的子节点。在此情况下,可以创建新的子节点6,其具有总存储空间标识 $6hw+5h+2w+30$ ,从而子节点2、新子节点6以及子节点3共具有可用存储空间 $15hw+9h+10w+40$ ,从而新节点8可以容纳数据D5。

[0109] 更进一步地,对于子节点3,由于之前存储有数据D3,因此子节点3需要被再次划分为孙节点7和孙节点4,其中孙节点4原先为子节点3的下级节点,而孙节点7为新创建的孙节点。

[0110] 更进一步地,如上文所述,优选地,同一个节点只能分配给相同的数据,因此节点2虽然具有大小为 $9hw+4h+6w+5$ 的可用存储空间,但由于节点2之前已经分配给数据D2,因此可以在节点2下创建新的节点5以作为节点2的子节点。需要理解的是,在存储器的物理空间上,节点2和节点5是一样的,但在树状结构中,二者是不同的节点。

[0111] 经过上述的创建过程,节点5能够提供大小 $9hw+4h+6w+5$ 的存储空间,节点6能够提供大小为 $6hw+4h+2w+30$ 的存储空间,并且节点7能够提供大小为 $h+2w+5$ 的存储空间,从而能够容纳大小为 $15hw+9h+10w+40$ 的数据D5。

[0112] 在上面针对图6b的描述中可以看出,同一个数据可以占用不同的节点,例如数据D5可以占用节点5,节点6和节点7,但相同的节点只能对应于一个数据,例如节点3已经分配给过数据D3,即使数据D3占用的空间已经被释放,但如果新的数据例如数据D4和数据D5需要占用数据D3原先占用的空间,则仍然需要将节点D3进一步划分为数节点4和节点7。上述的方案有利于保留整个节点与数据的对应关系。

[0113] 根据本公开的一个实施方式,创建新节点以使得可用存储空间能够容纳所述数据可以包括:响应于所述树状结构中所有节点的可用存储空间为零,创建新节点以使得所述新节点的可用存储空间能够容纳所述数据。

[0114] 在本实施方式中,由于现有的节点中已经没有任何可用存储空间,因此可以完全创建一个信息的节点来存储这些数据。图4a中的情形实际就是本实施方式的表现,因此这里将不再赘述。

[0115] 根据本公开的另一个实施方式中,创建新节点以使得可用存储空间能够容纳所述数据可以包括:响应于在所述树状结构的节点中未搜索到不小于所述数据大小的当前可用存储空间,并且下一空间分配请求包括的数据大小不大于所述当前可用存储空间,则创建新节点以使得所述新节点的可用存储空间能够容纳所述数据。

[0116] 在该实施方式中,可以将把空间分配请求的时序考虑进去。例如现有可用存储空间为 $6hw+4h+2w+30$ ,此时接收到第一个空间分配请求需要分配存储空间 $15hw+9h+10w+40$ ,则可以考虑该空间分配请求之后的第二个空间分配请求是否小于或等于 $6hw+4h+2w+30$ ,如果是,则可以首先创建新节点以分配给第一个空间分配请求,而将现有的可用存储空间预留给第二个空间分配请求。该实施方式对于周期性或者可预测的空间分配请求有利。

[0117] 根据本公开在图6c中建立新节点的方式,图4c所示的节点也可以根据图6c的方式来建立。

[0118] 图7a示出了根据本公开的一个实施方式的建立新节点的流程图,图 7b示出了根据本公开的一个实施方式的建立新节点的示意图,是图4c的另外一种实施方式。

[0119] 如图7a所示,根据本公开的一个实施方式创建新节点以使得可用存储空间能够容纳所述数据可以包括:在操作S710,响应于所述第一函数和第二函数均具有非公共部分,创建新节点;在操作S720,将其第二函数与所述第一函数具有公共函数的节点调整为所述新节点的下级节点;以及,在操作S730,在所述新节点下创建新下级节点以使得所述新节点的下级节点的可用存储空间能够容纳所述数据。

[0120] 在上文中,第一函数和第二函数均具有非公共部分是指第一函数与第二函数具有部分重叠和交叉部分,但彼此并不完全覆盖对方,这在上文中已经进行了介绍。新节点的总存储空间是两个函数涵盖的总的范围。具体而言,新节点的总存储空间是组成第一函数和第二函数的各个项中大者之和。以节点4b为例,表示节点1的总存储空间的第二函数是 $10hw+5h+5w+20$ ,其具有四个项,分别为 $10hw$ , $5h$ , $5w$ 和 $20$ ;空间分配请求中表示数据大小的第一函数为 $8hw+10h+10w+100$ ,其四个项分别为 $8hw$ , $10h$ , $10w$ 和 $100$ ,因此要申请的新节点的总存储空间为 $10hw+10h+10w+100$ ,新节点的总存储空间减去第一函数就是节点1中剩余的可用存储空间(例如为 $2hw$ ),新节点的总存储空间减去第二函数就是需要为该数据重新申请的新子节点的空间大小(例如为 $5h+5w+80$ )。

[0121] 由此,如图7b所示,可以创建一个新节点6,其可总存储空间为 $10hw+10h+10w+100$ ,该节点6与节点2和节点3处于树状结构的同一年级;将节点1移动到节点6下作为节点6的子节点,其中节点1的总存储空间为 $10hw+5h+5w+20$ ,可用存储空间为 $2hw$ ;创建新的节点5作为节点6 的子节点,其中节点5的总存储空间为 $5h+5w+80$ ,可用存储空间为 $0$ ;节点4仍然为节点1的子节点。在图7b,需要理解的是,节点5和节点4 由于均用于存储数据B,因此优选地这两个节点紧邻,从而使得同样的数据能够存储在连续的空间中,减少空间寻址时的频繁

跳变。

[0122] 可以理解的是,在树状结构中具有可用存储空间的节点不足以存储数据需要创建新节点的情况下,创建的新节点的下级节点的存储空间越小越好,这能够最大限度的复用树状结构的可用存储空间,有利于减少空间碎片的发生。可选地,在树状结构中具有可用存储空间的节点不足以存储数据时,结合图4d所描述的第五种和第六种方案同样适用。在此情况下,必然会创建新的下级节点,与第五种和第六种方案相似,可以选择其中一个节点来容纳一部分数据,然后创建新节点的下级节点来容纳剩余部分的数据。所创建的新节点的下级节点可以与具有可用存储空间的节点构成连续的空间,从而减少数据存取时的空间跳变。

[0123] 当建立了树状结构之后并且接收到一个新的空间分配请求时,可以遍历该树状结构以找到期望的可用存储空间。根据本公开的一个实施方式,在遍历该树状结构时,可以采用深度优先(depth first)的策略,即检索完一个上级节点以及该上级节点的所有下级节点之后,再检索与该上级节点同级的另外一个节点。

[0124] 以图6c为例,例如接收到一个空间分配请求,那么首先要查找节点1,当节点1中未找到足够的可用存储空间时,查找节点8,然后依次查找节点2和节点5;查找节点6;查找节点3、节点7以及节点4。深度优先的方式有利于尽量将同一个数据存储在连续的空间中,减少运算过程中空间寻址的跳变。

[0125] 根据本公开的一个实施方式,进一步包括:调整所述节点的位置,以使得所述多个存储块的空间是连续的。

[0126] 需要理解的是,节点实际上对应是虚拟的存储空间,因此希望同一块数据能够存储在连续的空间中,以减少运算过程中空间寻址的跳变,提升运算效率。

[0127] 如图6c所示,用于存储数据D5的节点6插在了节点2和节点3之间,而节点7的建立紧邻节点6,由此,数据D5可以连续地存储在节点5,节点6和节点7紧邻的连续空间中。节点6也可以设置在节点3之后,节点7设置在节点4之后,但这样的方式容易引起空间跳变。

[0128] 由此,为了保证空间连续,根据本公开的一个实施方式,进一步包括:固定节点在所述树状结构中的排列顺序。

[0129] 固定节点的顺序可以通过在节点的上级节点中做标记。例如,如图6c 所示,可以在节点8中打上标记“ordered”,那么该节点8下的所有子节点的排列顺序将固定下来。又例如,可以给节点3打上标记“ordered”,那么该节点3下的节点7和节点4的排列顺序将被固定下来。

[0130] 固定节点的排列顺序有助于使得同一数据存储在连续的空间中,减少了空间寻址的跳变。

[0131] 与现有技术不同的是,在上面的过程中,仅仅是分配存储空间,而并未为数据分配地址,因此空间分配和地址分配是分离的;而在现有技术中,分配地址和分配存储空间是同时的,这也是本公开的技术方案与现有技术方案之间的一个不同之处。

[0132] 图8示出了根据本公开的一个实施方式的为存储空间分配地址的示意图。为了方便理解,图8中每个节点的总存储空间大小以具体的数字来表示。

[0133] 如图8所示的树状结构包括节点0、节点1、节点2、节点3、节点4 和节点5,节点0为根节点,其总存储空间为31,可用存储空间为0,存储属性表示为31(0);节点1,节点2,节点3

和节点4是节点0的子节点,其存储属性分别表示为5(5),6(6),10(0)和11(0);节点5是节点3的子节点,其存储属性表示为10(0)。

[0134] 在为这些节点分配地址时,按照宽度优先(width-first)的策略,即首先为同一级的节点分配地址,再更具体地为每个节点的下级节点分配地址。

[0135] 如图8所示,节点0的起始地址为0x0000,其偏移量可以为31;那么作为节点0的第一个子节点,节点1的起始地址也可以为0x0000,其偏移量为5,因此其结束地址为0x0004。类似地,节点2的起始地址可以为0x0005,偏移量为6;节点3的起始地址可以为0x000B,偏移量为10;而节点4的起始地址为0x0015,偏移量为11;作为节点3的子节点,节点5的起始地址也为0x000B,偏移量为10。

[0136] 如果将上面具体的数字换做函数表示,假设节点0的总存储空间大小为 $20hw + 15h + 26w + 113$ 并且其实地址为0x0000,那么节点0的地址范围就是 $0x0000 + (20hw + 15h + 26w + 113) - 1$ 。

[0137] 可以看出,通过上述的安排,相同数据的空间可以保持连续,从而也可以分配连续的地址,这极大地提升了存储器中的空间利用率,减少了碎片的发生;此外,通过本公开的技术方案,还可以保持数据存储的连续性,从而避免了寻址过程中的跳变,从而提升数据吞吐的速度;另外一个有益的效果是,由于所有的节点信息都得以保留,因此保留了空间分配和释放过程中数据存储的所有记录,便于后续的数据追踪和存储记录的保留。

[0138] 本公开的方案尤其适用于存储量较大的存储器,例如片外存储器,硬盘等。

[0139] 图9示出了根据本公开的一个实施方式的对存储空间进行管理的方法,包括:在操作S910,形成多个存储块,每个存储块具有总存储空间标识和可用存储空间标识,其中总存储空间标识用于指示所述存储块的总存储空间,所述可用存储空间标识用于指示所述存储块的可用存储空间;以及,在操作S920,将所述多个存储块作为节点形成为树状结构,所述树状结构包括多级节点,其中上级节点的总存储空间为下级节点的总存储空间之和。

[0140] 图10示出了根据本公开的一个实施方式的在存储器中为数据分配存储空间的方法,包括:在操作S1010,接收针对存储器中存储空间的空间分配请求,所述分配请求携带以第一函数表示的数据大小;在操作S1020,形成多个存储块,每个存储块具有总存储空间标识和可用存储空间标识,其中总存储空间标识用于以总空间函数来指示所述存储块的总存储空间,所述可用存储空间标识用于以第二函数来指示所述存储块的可用存储空间;在操作S1030,将所述多个存储块作为节点形成为树状结构,所述树状结构包括多级节点,其中上级节点的总存储空间为下级节点的总存储空间之和;以及,在操作S1040,根据接收到的所述空间分配请求,根据所述树状结构来为数据分配目标存储空间。

[0141] 上述图9和图10中各个操作的具体实施方式已经根据图2-图8进行了详细解释,因此这里将不再赘述。

[0142] 本公开还提供了一种种电子设备,包括:一个或多个处理器;以及存储器,所述存储器中存储有计算机可执行指令,当所述计算机可执行指令由所述一个或多个处理器运行时,使得所述电子设备执行如上所述的方法。

[0143] 本公开还提供了一种计算机可读存储介质,包括计算机可执行指令,当所述计算机可执行指令由一个或多个处理器运行时,执行如上所述的方法。

[0144] 本公开的技术方案可应用于人工智能领域,实现在主机中、服务器中,或者实现为

或者实现在人工智能芯片中。该芯片可以单独存在,也可以包含在计算装置中。

[0145] 图11示出了一种组合处理装置1100,其包括上述的计算装置1102,通用互联接口1104,和其他处理装置1106。根据本公开的计算装置与其他处理装置进行交互,共同完成用户指定的操作。图11为组合处理装置的示意图。

[0146] 其他处理装置,包括中央处理器CPU、图形处理器GPU、神经网络处理器等通用/专用处理器中的一种或以上的处理器类型。其他处理装置所包括的处理器数量不做限制。其他处理装置作为机器学习计算装置与外部数据和控制的接口,包括数据搬运,完成对本机器学习计算装置的开启、停止等基本控制;其他处理装置也可以和机器学习计算装置协作共同完成计算任务。

[0147] 通用互联接口,用于在计算装置(包括例如机器学习计算装置)与其他处理装置间传输数据和控制指令。该计算装置从其他处理装置中获取所需的输入数据,写入该计算装置片上的存储装置;可以从其他处理装置中获取控制指令,写入计算装置片上的控制缓存;也可以读取计算装置的存储模块中的数据并传输给其他处理装置。

[0148] 可选的,该结构还可以包括存储装置1108,存储装置分别与所述计算装置和所述其他处理装置连接。存储装置用于保存在所述计算装置和所述其他处理装置的数据,尤其适用于所需要计算的数据在本计算装置或其他处理装置的内部存储中无法全部保存的数据。

[0149] 该组合处理装置可以作为手机、机器人、无人机、视频监控设备等设备的SOC片上系统,有效降低控制部分的核心面积,提高处理速度,降低整体功耗。此情况时,该组合处理装置的通用互联接口与设备的某些部件相连接。某些部件譬如摄像头,显示器,鼠标,键盘,网卡,wifi接口。

[0150] 在一些实施例里,本披露还公开了一种板卡,其包括了上述芯片封装结构。参阅图12,其提供了一种示例性的板卡,上述板卡除了包括上述芯片1202以外,还可以包括其他的配套部件,该配套部件包括但不限于:存储器件1204、接口装置1206和控制器件1208。

[0151] 所述存储器件与所述芯片封装结构内的芯片通过总线连接,用于存储数据。所述存储器件可以包括多组存储单元1210。每一组所述存储单元与所述芯片通过总线连接。可以理解,每一组所述存储单元可以是DDR SDRAM(英文:Double Data Rate SDRAM,双倍速率同步动态随机存储器)。

[0152] DDR不需要提高时钟频率就能加倍提高SDRAM的速度。DDR允许在时钟脉冲的上升沿和下降沿读出数据。DDR的速度是标准SDRAM的两倍。在一个实施例中,所述存储装置可以包括4组所述存储单元。每一组所述存储单元可以包括多个DDR4颗粒(芯片)。在一个实施例中,所述芯片内部可以包括4个72位DDR4控制器,上述72位DDR4控制器中 64bit用于传输数据,8bit用于ECC校验。在一个实施例中,每一组所述存储单元包括多个并联设置的双倍速率同步动态随机存储器。DDR在一个时钟周期内可以传输两次数据。在所述芯片中设置控制DDR的控制器,用于对每个所述存储单元的数据传输与数据存储的控制。

[0153] 所述接口装置与所述芯片封装结构内的芯片电连接。所述接口装置用于实现所述芯片与外部设备1212(例如服务器或计算机)之间的数据传输。例如在一个实施例中,所述接口装置可以为标准PCIE接口。比如,待处理的数据由服务器通过标准PCIE接口传递至所述芯片,实现数据转移。在另一个实施例中,所述接口装置还可以是其他的接口,本披露并

不限制上述其他的接口的具体表现形式,所述接口单元能够实现转接功能即可。另外,所述芯片的计算结果仍由所述接口装置传送回外部设备(例如服务器)。

[0154] 所述控制器件与所述芯片电连接。所述控制器件用于对所述芯片的状态进行监控。具体的,所述芯片与所述控制器件可以通过SPI接口电连接。所述控制器件可以包括单片机(Micro Controller Unit, MCU)。如所述芯片可以包括多个处理芯片、多个处理核或多个处理电路,可以带动多个负载。因此,所述芯片可以处于多负载和轻负载等不同的工作状态。通过所述控制装置可以实现对所述芯片中多个处理芯片、多个处理和/或多个处理电路的工作状态的调控。

[0155] 在一些实施例里,本披露还公开了一种电子设备或装置,其包括了上述板卡。

[0156] 电子设备或装置包括数据处理装置、机器人、电脑、打印机、扫描仪、平板电脑、智能终端、手机、行车记录仪、导航仪、传感器、摄像头、服务器、云端服务器、相机、摄像机、投影仪、手表、耳机、移动存储、可穿戴设备、交通工具、家用电器、和/或医疗设备。

[0157] 所述交通工具包括飞机、轮船和/或车辆;所述家用电器包括电视、空调、微波炉、冰箱、电饭煲、加湿器、洗衣机、电灯、燃气灶、油烟机;所述医疗设备包括核磁共振仪、B超仪和/或心电图仪。

[0158] 需要说明的是,对于前述的各方法实施例,为了简单描述,故将其都表述为一系列的动作组合,但是本领域技术人员应该知悉,本披露并不受所描述的动作顺序的限制,因为依据本披露,某些步骤可以采用其他顺序或者同时进行。其次,本领域技术人员也应该知悉,说明书中所描述的实施例均属于可选实施例,所涉及的动作和模块并不一定是本披露所必须的。

[0159] 在上述实施例中,对各个实施例的描述都各有侧重,某个实施例中没有详述的部分,可以参见其他实施例的相关描述。

[0160] 在本披露所提供的几个实施例中,应该理解到,所披露的装置,可通过其它的方式实现。例如,以上所描述的装置实施例仅仅是示意性的,例如所述单元的划分,仅仅为一种逻辑功能划分,实际实现时可以有另外的划分方式,例如多个单元或组件可以结合或者可以集成到另一个系统,或一些特征可以忽略,或不执行。另一点,所显示或讨论的相互之间的耦合或直接耦合或通信连接可以是通过一些接口,装置或单元的间接耦合或通信连接,可以是电性、光学、声学、磁性或其它的形式。

[0161] 所述作为分离部件说明的单元可以是或者也可以不是物理上分开的,作为单元显示的部件可以是或者也可以不是物理单元,即可以位于一个地方,或者也可以分布到多个网络单元上。可以根据实际的需要选择其中的部分或者全部单元来实现本实施例方案的目的。

[0162] 另外,在本披露各个实施例中的各功能单元可以集成在一个处理单元中,也可以是各个单元单独物理存在,也可以两个或两个以上单元集成在一个单元中。上述集成的单元既可以采用硬件的形式实现,也可以采用软件程序模块的形式实现。

[0163] 所述集成的单元如果以软件程序模块的形式实现并作为独立的产品销售或使用时,可以存储在一个计算机可读取存储器中。基于这样的理解,当本披露的技术方案可以以软件产品的形式体现出来,该计算机软件产品存储在一个存储器中,包括若干指令用以使得一台计算机设备(可为个人计算机、服务器或者网络设备等)执行本披露各个实施例所述

方法的全部或部分步骤。而前述的存储器包括:U盘、只读存储器(ROM, Read-Only Memory)、随机存取存储器(RAM, Random Access Memory)、移动硬盘、磁碟或者光盘等各种可以存储程序代码的介质。

[0164] 以上对本披露实施例进行了详细介绍,本文中应用了具体个例对本披露的原理及实施方式进行了阐述,以上实施例的说明只是用于帮助理解本披露的方法及其核心思想;同时,对于本领域的一般技术人员,依据本披露的思想,在具体实施方式及应用范围上均会有改变之处,综上所述,本说明书内容不应理解为对本披露的限制。

[0165] 依据以下条款可更好地理解前述内容:

[0166] 条款1.一种在存储器中为数据分配存储地址的方法,包括:

[0167] 接收针对存储器中存储空间的空间分配请求,所述分配请求携带以第一函数表示的数据大小;

[0168] 根据接收到的所述空间分配请求来为数据分配目标存储空间;以及

[0169] 基于所分配的目标存储空间,为所述数据分配存储地址。

[0170] 条款2.根据条款1所述的方法,进一步包括:

[0171] 形成多个存储块,每个存储块具有总存储空间标识和可用存储空间标识,其中总存储空间标识用于以总空间函数来指示所述存储块的总存储空间,所述可用存储空间标识用于以第二函数来指示所述存储块的可用存储空间;

[0172] 将所述多个存储块作为节点形成为树状结构,所述树状结构包括多级节点,其中上级节点的总存储空间为下级节点的总存储空间之和。

[0173] 条款3.根据条款2所述的方法,其中,根据接收到的所述空间分配请求来为数据分配目标存储空间包括:

[0174] 在所述树状结构的节点中检索可用存储空间;

[0175] 响应于在所述树状结构的节点中搜索到其第二函数与所述第一函数具有公共函数的可用存储空间,将所述公共函数所指示的存储空间作为目标存储空间。

[0176] 条款4.根据条款3所述的方法,其中,响应于在所述树状结构的节点中搜索到其第二函数与所述第一函数具有最大公共函数的可用存储空间,将所述公共函数所指示的存储空间作为目标存储空间。

[0177] 条款5.根据条款3所述的方法,其中,重复在所述树状结构的节点中检索可用存储空间,直到检索不到可用存储空间为止。

[0178] 条款6.根据条款3所述的方法,其中,根据接收到的所述空间分配请求来为数据分配目标存储空间包括:

[0179] 响应于在所述树状结构的节点中未搜索到不小于所述数据大小的可用存储空间,创建新节点以使得可用存储空间能够容纳所述数据。

[0180] 条款7.根据条款6所述的方法,其中,创建新节点以使得可用存储空间能够容纳所述数据包括:

[0181] 将具有可用存储空间的节点调整为所述新节点的下级节点;

[0182] 在所述新节点下创建新下级节点以使得所述新节点的下级节点的可用存储空间能够容纳所述数据;或者

[0183] 创建新节点以使得可用存储空间能够容纳所述数据包括:

- [0184] 响应于所述树状结构中所有节点的可用存储空间为零,创建新节点以使得所述新节点的可用存储空间能够容纳所述数据;或者
- [0185] 创建新节点以使得可用存储空间能够容纳所述数据包括:
- [0186] 响应于在所述树状结构的节点中未搜索到不小于所述数据大小的当前可用存储空间,并且下一空间分配请求包括的数据大小不大于所述当前可用存储空间,则创建新节点以使得所述新节点的可用存储空间能够容纳所述数据。
- [0187] 条款8.根据条款6所述的方法,其中,创建新节点以使得可用存储空间能够容纳所述数据包括:
- [0188] 响应于所述第一函数和第二函数均具有非公共部分,创建新节点;
- [0189] 将其第二函数与所述第一函数具有公共函数的节点调整为所述新节点的下级节点;以及,
- [0190] 在所述新节点下创建新下级节点以使得所述新节点的下级节点的可用存储空间能够容纳所述数据。
- [0191] 条款9.根据条款3-8中任意一项所述的方法,其中,在所述树状结构的节点中检索可用存储空间包括:
- [0192] 在检索完一个上级节点以及该上级节点的所有下级节点之后,再检索与该上级节点同级的另外一个节点。
- [0193] 条款10.根据条款2-9中任意一项所述的方法,进一步包括:
- [0194] 调整所述节点的位置,以使得所述多个存储块的空间是连续的。
- [0195] 条款11.根据条款2-10中任意一项所述的方法,进一步包括:
- [0196] 固定节点在所述树状结构中的排列顺序。
- [0197] 条款12.根据条款2-11中任意一项所述的方法,进一步包括:
- [0198] 响应于接收到空间释放请求,释放相应的节点;
- [0199] 更新被释放节点的可用存储空间标识符。
- [0200] 条款13.根据条款2-12中任意一项所述的方法,基于所分配的目标存储空间,为所述数据分配存储地址包括:
- [0201] 在为一个节点分配存储地址之后,依次为同级的其他节点分配地址,其中每个节点的地址涵盖该节点所有下级节点的地址。
- [0202] 条款14.一种对存储空间进行管理的方法,包括:
- [0203] 形成多个存储块,每个存储块具有总存储空间标识和可用存储空间标识,其中总存储空间标识用于以总空间函数来指示所述存储块的总存储空间,所述可用存储空间标识用于以第二函数来指示所述存储块的可用存储空间;
- [0204] 将所述多个存储块作为节点形成为树状结构,所述树状结构包括多级节点,其中上级节点的总存储空间为下级节点的总存储空间之和。
- [0205] 条款15.一种在存储器中为数据分配存储空间的方法,包括:
- [0206] 接收针对存储器中存储空间的空间分配请求,所述分配请求携带以第一函数表示的数据大小;
- [0207] 形成多个存储块,每个存储块具有总存储空间标识和可用存储空间标识,其中总存储空间标识用于以总空间函数来指示所述存储块的总存储空间,所述可用存储空间标识

用于以第二函数来指示所述存储块的可用存储空间；

[0208] 将所述多个存储块作为节点形成为树状结构，所述树状结构包括多级节点，其中上级节点的总存储空间为下级节点的总存储空间之和；

[0209] 根据接收到的所述空间分配请求，根据所述树状结构来为数据分配目标存储空间。

[0210] 条款16.根据条款14或15所述的方法，进一步包括：

[0211] 响应于接收到为数据分配目标存储空间的空间分配请求，在所述树状结构的节点中检索可用存储空间；

[0212] 响应于在所述树状结构的节点中搜索到其第二函数与所述第一函数具有公共函数的可用存储空间，将所述公共函数所指示的存储空间作为目标存储空间。

[0213] 条款17.根据条款16所述的方法，其中，响应于在所述树状结构的节点中搜索到其第二函数与所述第一函数具有最大公共函数的可用存储空间，将所述公共函数所指示的存储空间作为目标存储空间。

[0214] 条款18.根据条款16所述的方法，其中，重复在所述树状结构的节点中检索可用存储空间，直到检索不到可用存储空间为止。

[0215] 条款19.根据条款16所述的方法，其中，响应于在所述树状结构的节点中未搜索到不小于所述数据大小的可用存储空间，创建新节点以使得可用存储空间能够容纳所述数据。

[0216] 条款20.根据条款19所述的方法，其中，创建新节点以使得可用存储空间能够容纳所述数据包括：

[0217] 将具有可用存储空间的节点调整为所述新节点的下级节点；

[0218] 在所述新节点下创建新下级节点以使得所述新节点的下级节点的可用存储空间能够容纳所述数据；或者

[0219] 创建新节点以使得可用存储空间能够容纳所述数据包括：

[0220] 响应于所述树状结构中所有节点的可用存储空间为零，创建新节点以使得所述新节点的可用存储空间能够容纳所述数据；或者

[0221] 创建新节点以使得可用存储空间能够容纳所述数据包括：

[0222] 响应于在所述树状结构的节点中未搜索到不小于所述数据大小的当前可用存储空间，并且下一空间分配请求包括的数据大小不大于所述当前可用存储空间，则创建新节点以使得所述新节点的可用存储空间能够容纳所述数据。

[0223] 条款21.根据条款16-20中任意一项所述的方法，其中，在所述树状结构的节点中检索可用存储空间包括：

[0224] 在检索完一个上级节点以及该上级节点的所有下级节点之后，再检索与该上级节点同级的另外一个节点。

[0225] 条款22.根据条款14-21中任意一项所述的方法，进一步包括：

[0226] 调整所述节点的位置，以使得所述多个存储块的空间是连续的。

[0227] 条款23.根据条款14-22中任意一项所述的方法，进一步包括：

[0228] 固定节点在所述树状结构的中的排列顺序。

[0229] 条款24.一种电子设备，包括：

[0230] 一个或多个处理器；以及

[0231] 存储器，所述存储器中存储有计算机可执行指令，当所述计算机可执行指令由所述一个或多个处理器运行时，使得所述电子设备执行如条款 1-23 中任意一项所述的方法。

[0232] 条款25.一种计算机可读存储介质，包括计算机可执行指令，当所述计算机可执行指令由一个或多个处理器运行时，执行如条款1-23中任意一项所述的方法。

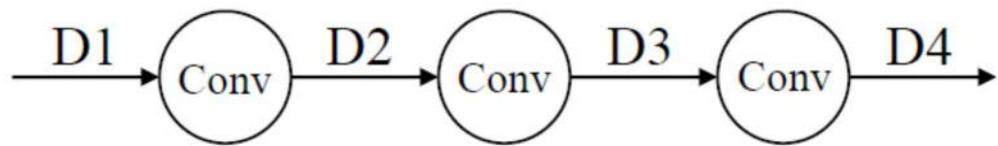


图1a

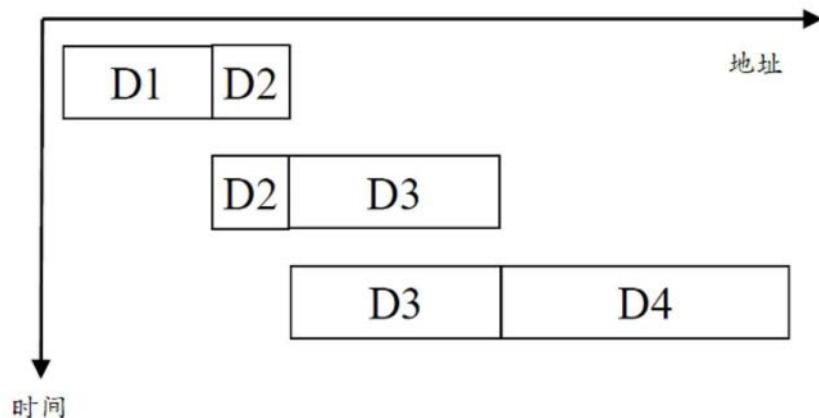


图1b

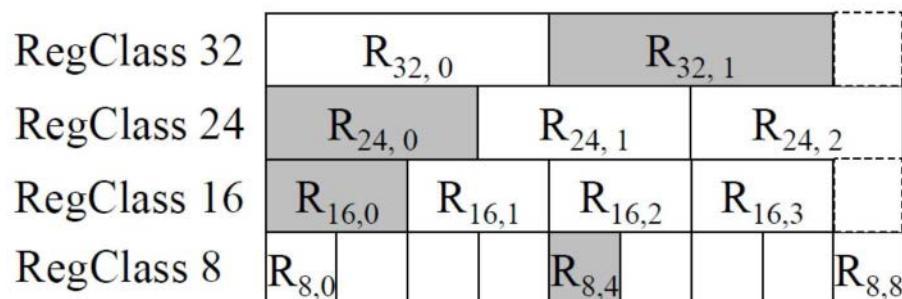


图1c

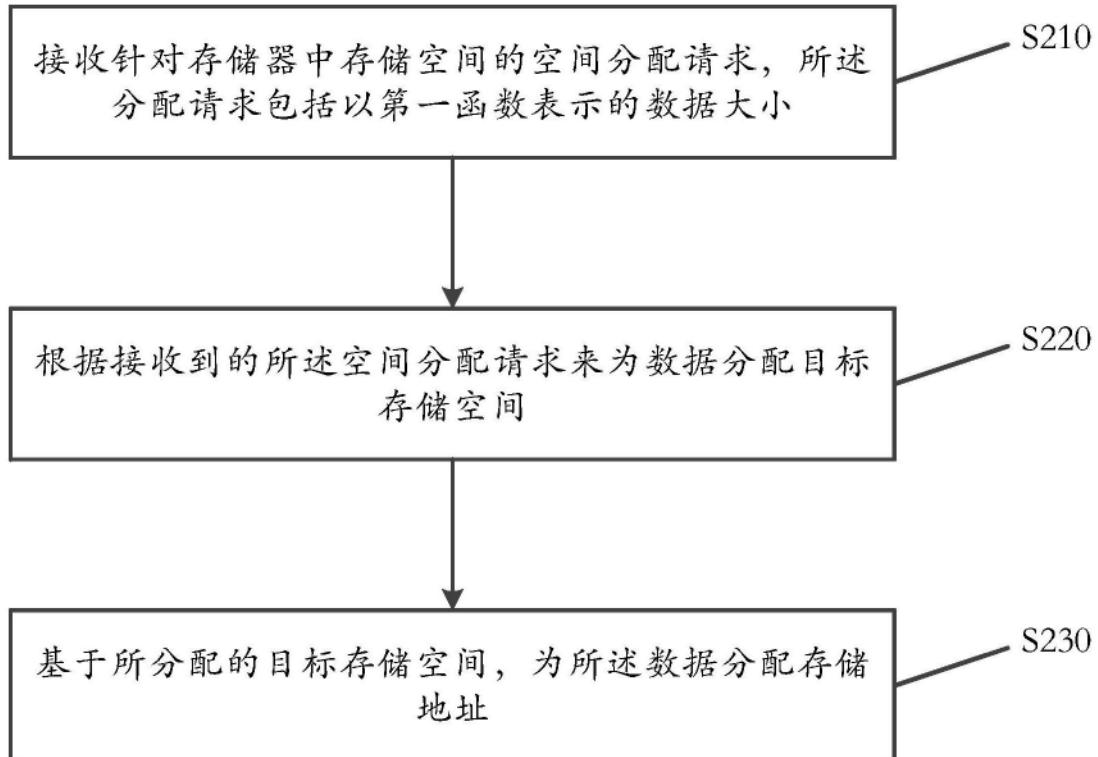


图2

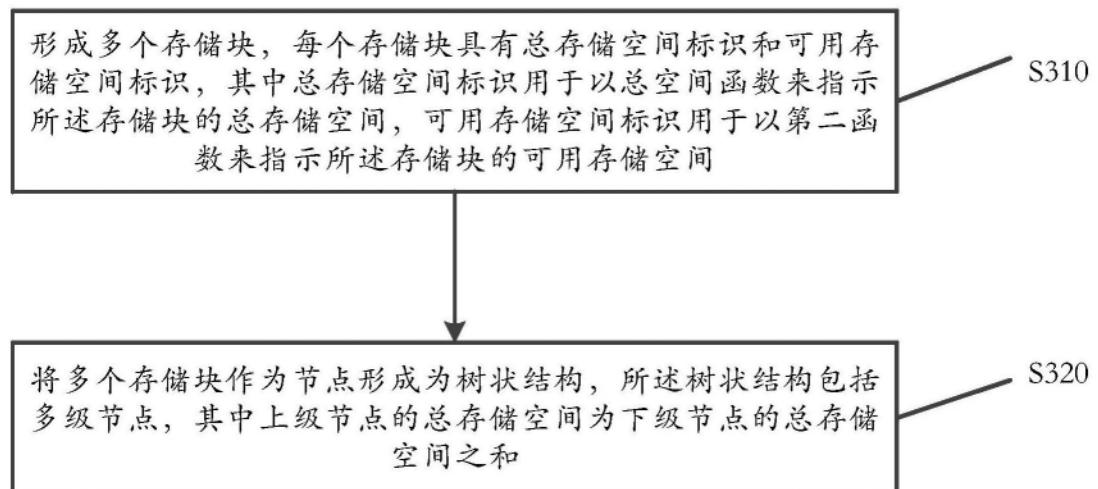


图3

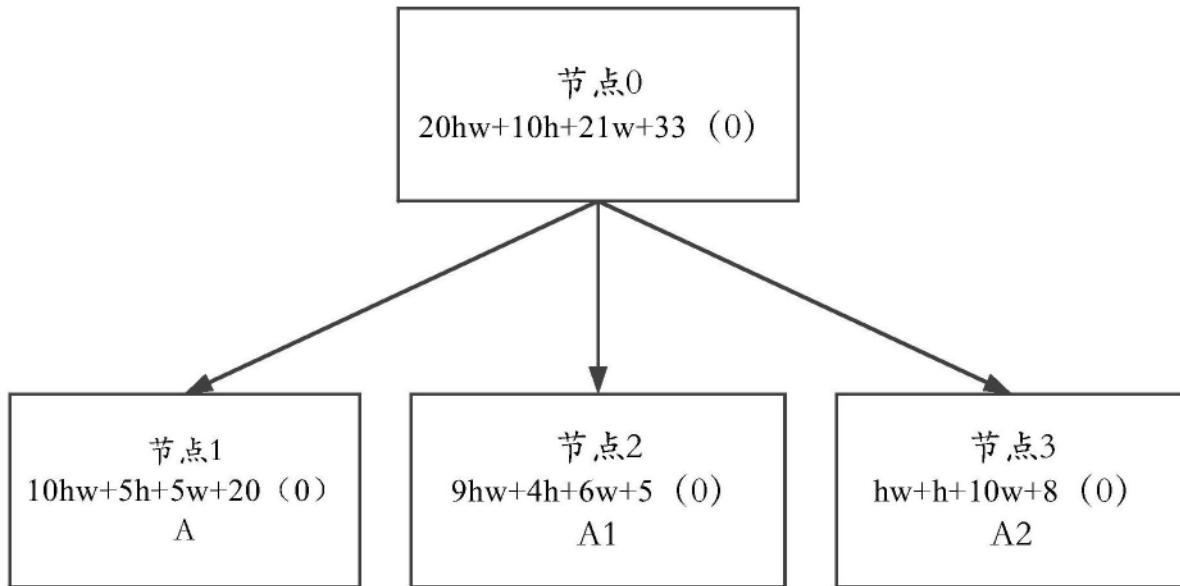


图4a

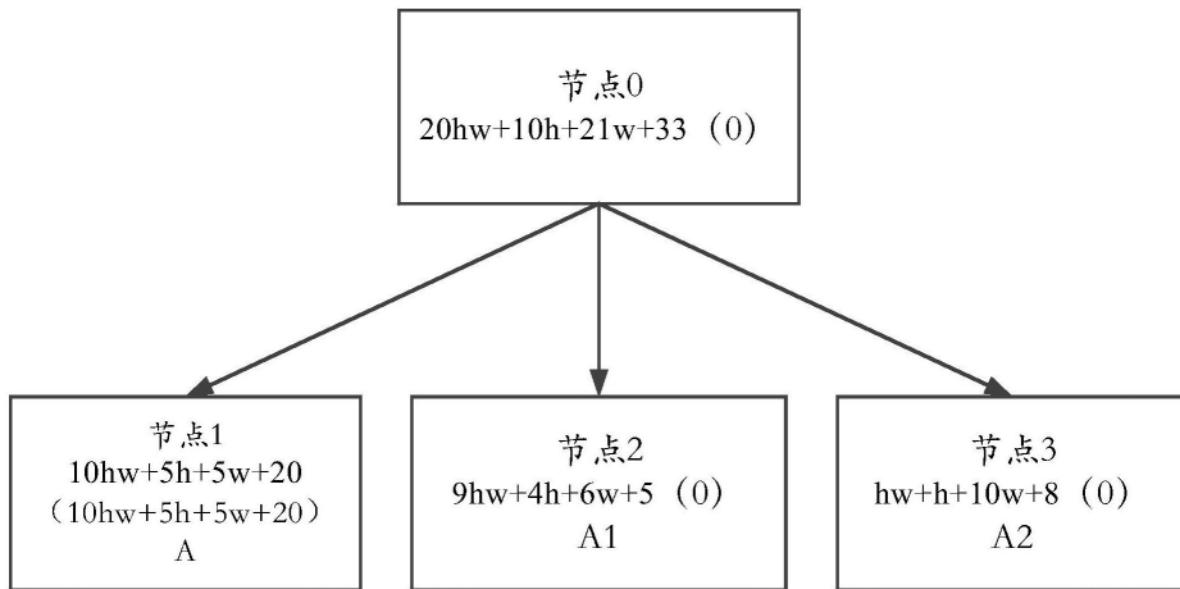


图4b

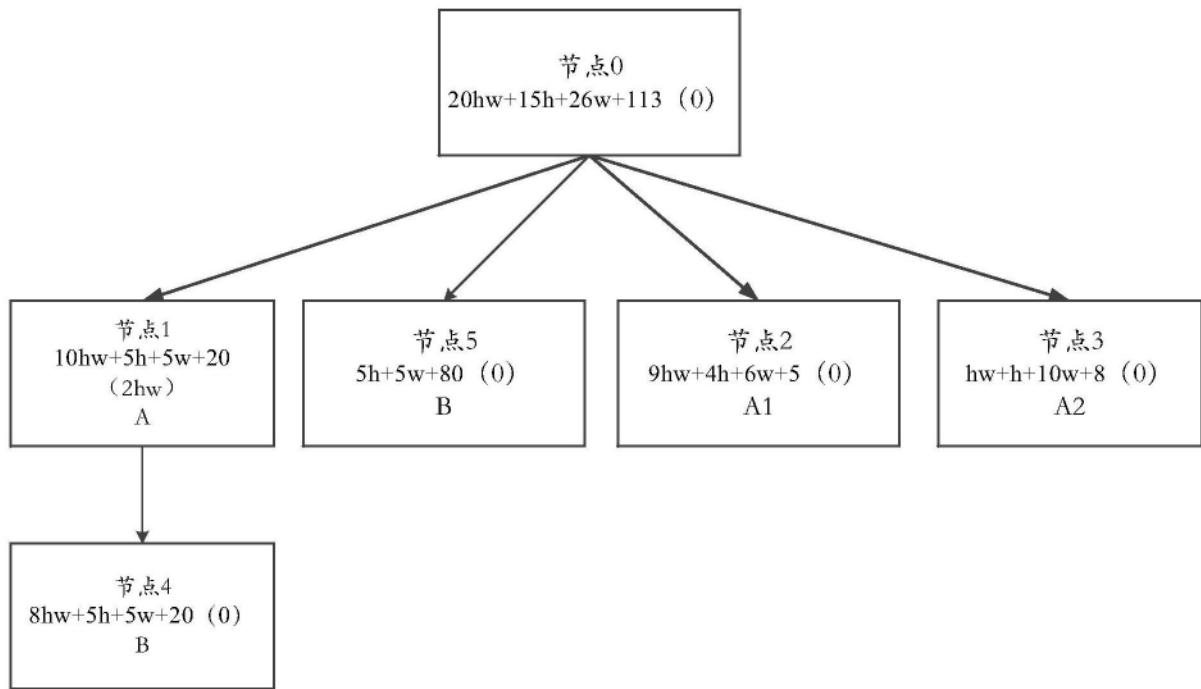


图4c

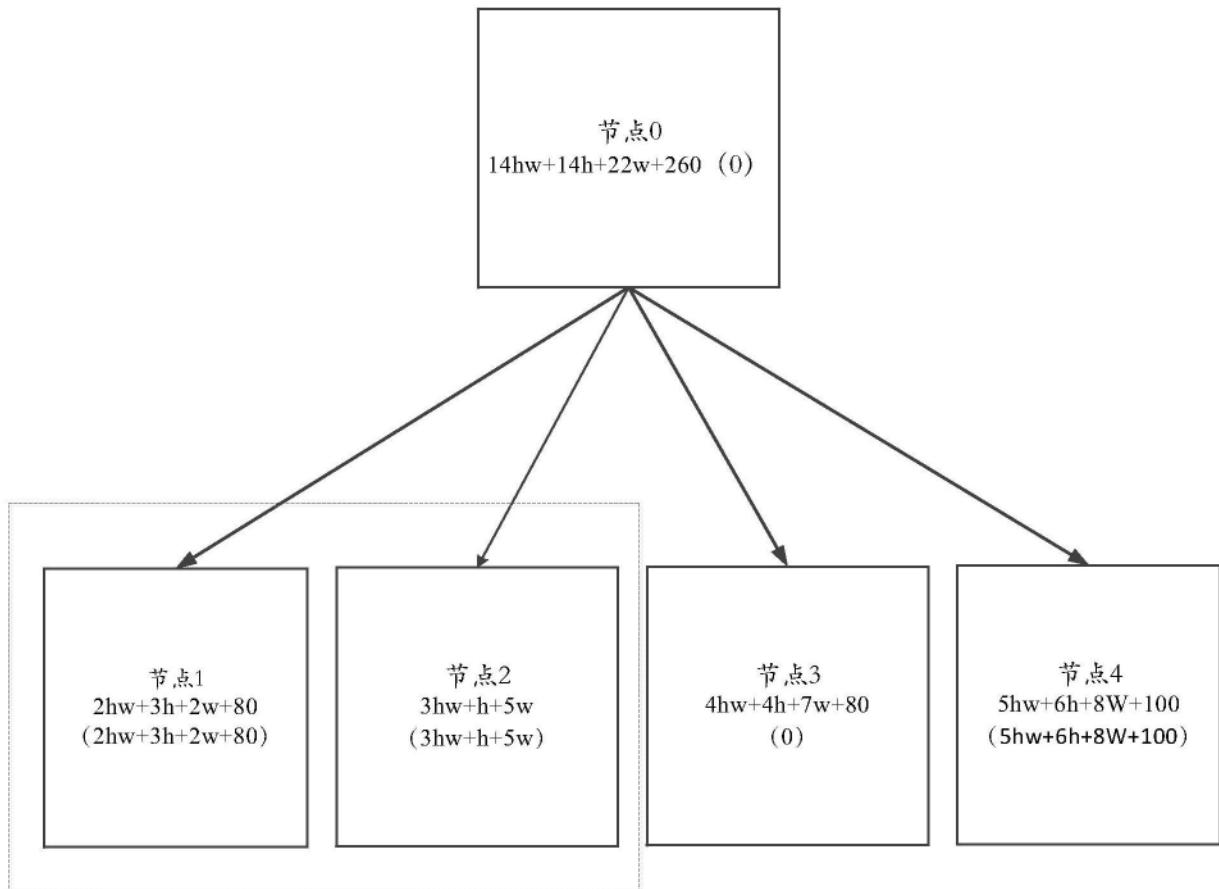


图4d

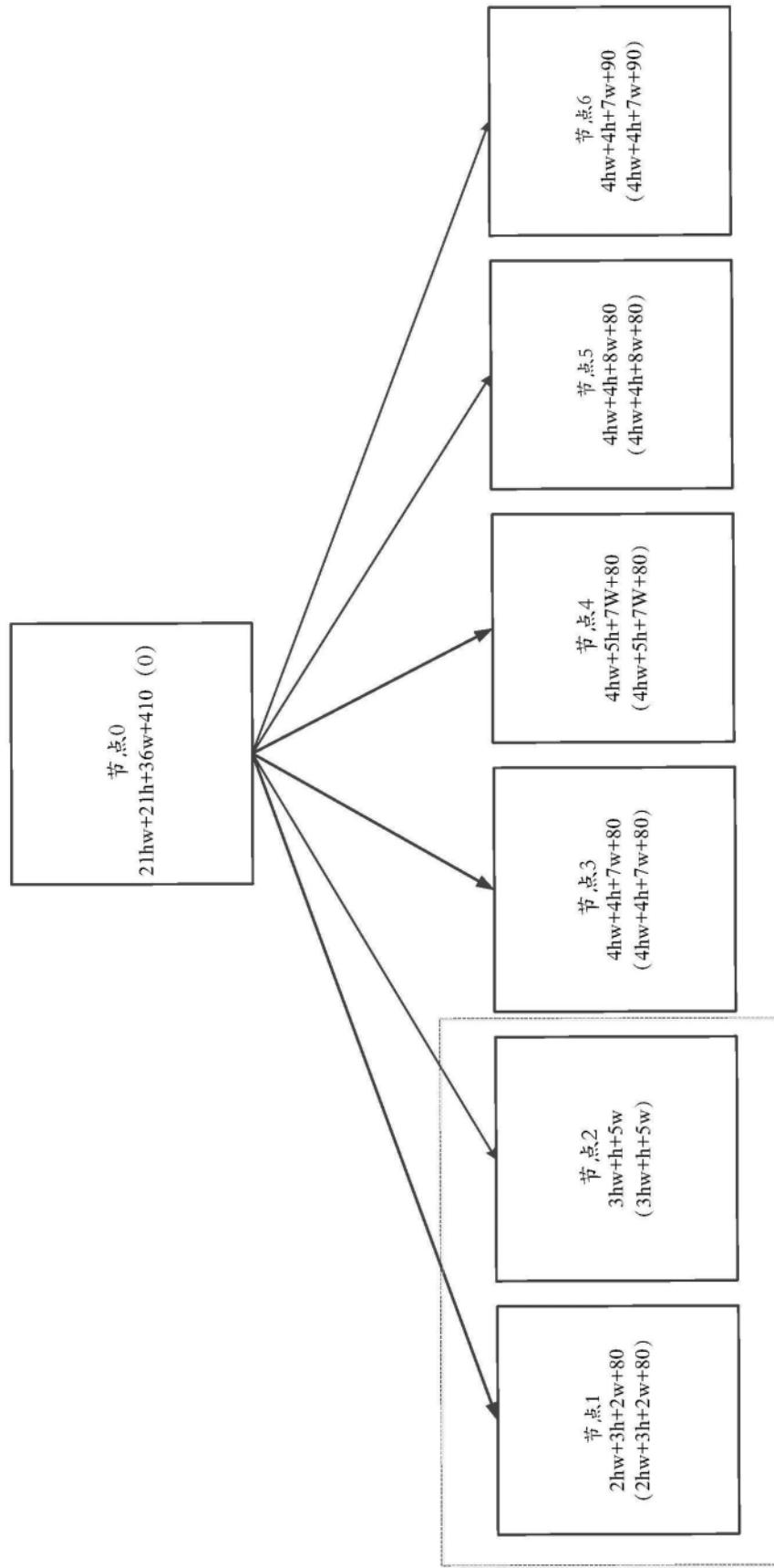


图5

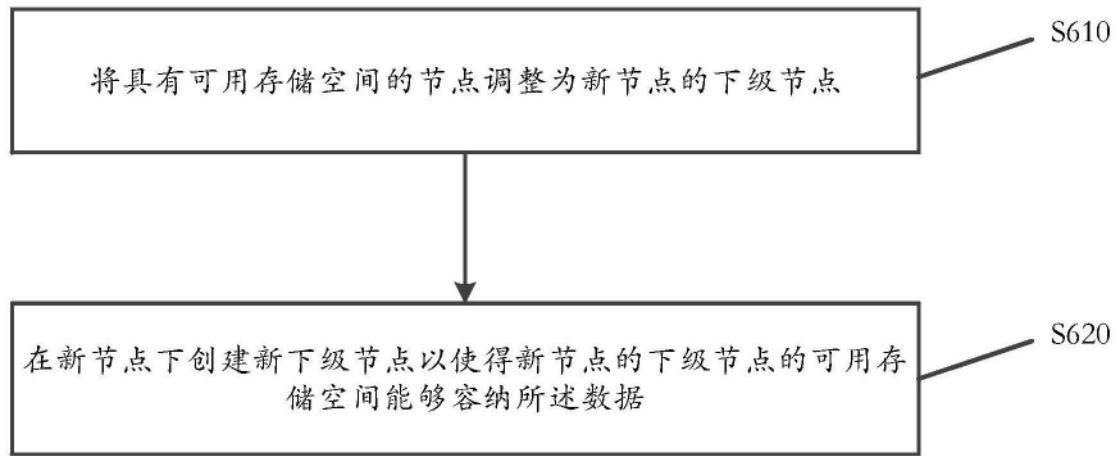


图6a

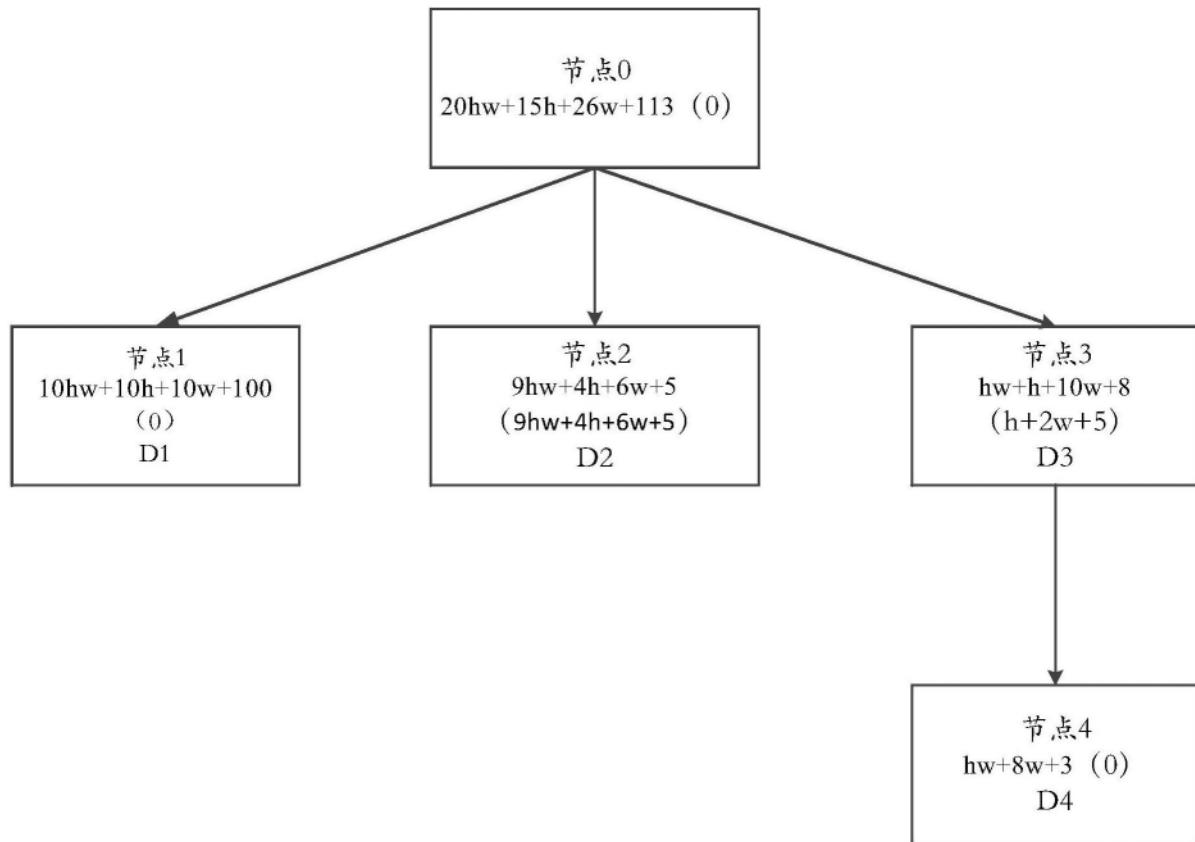


图6b

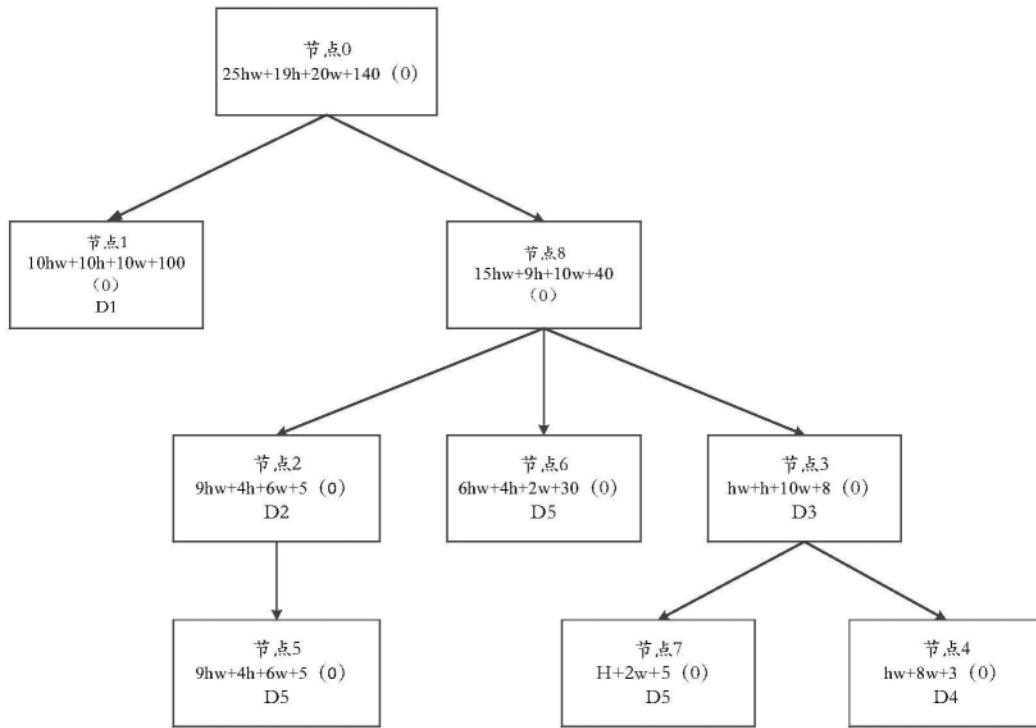


图6c

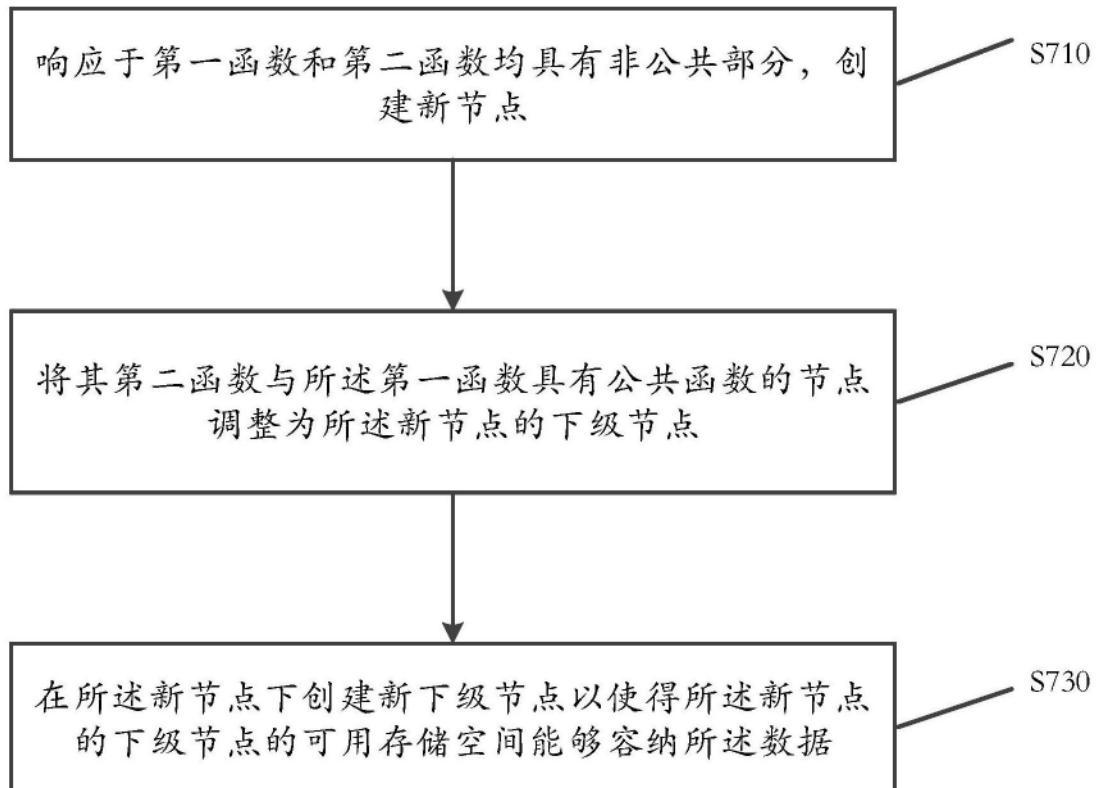


图7a

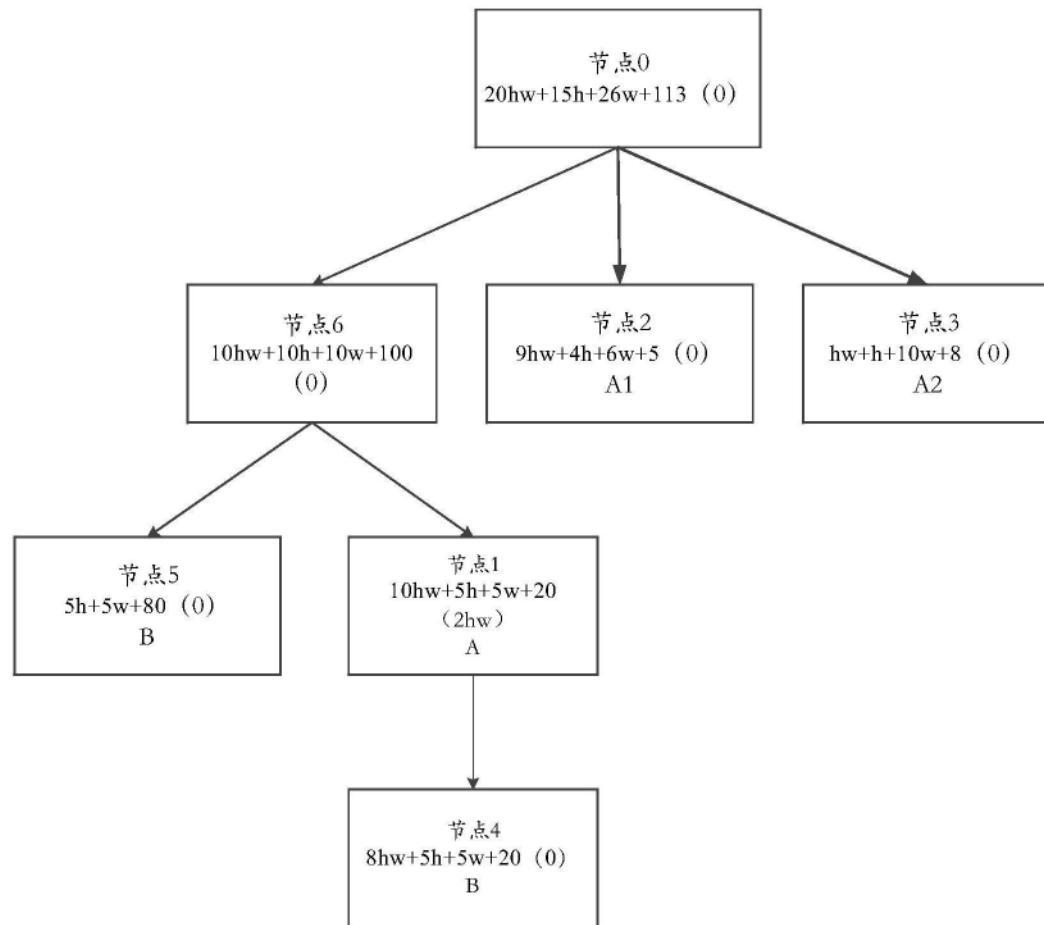


图7b

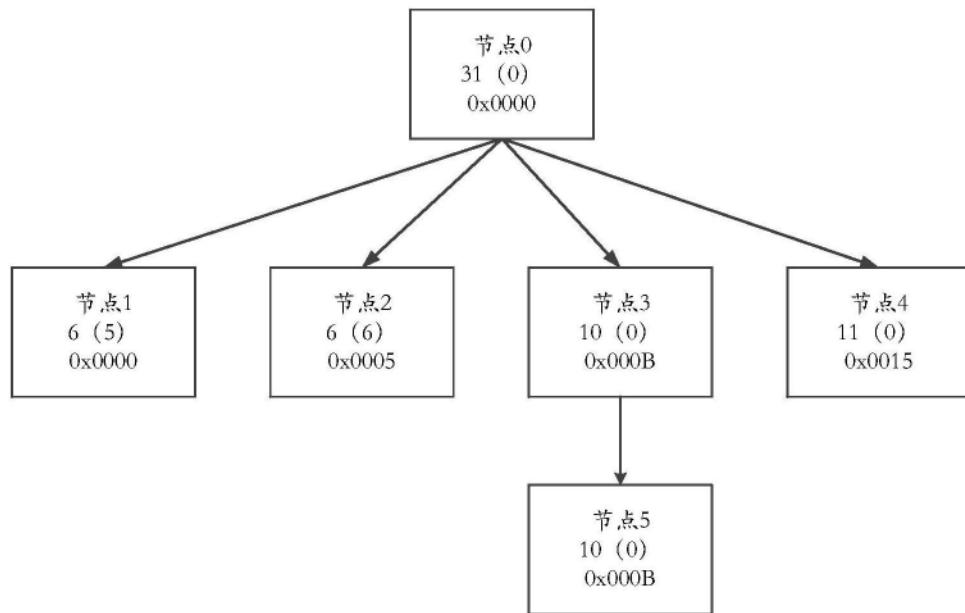


图8

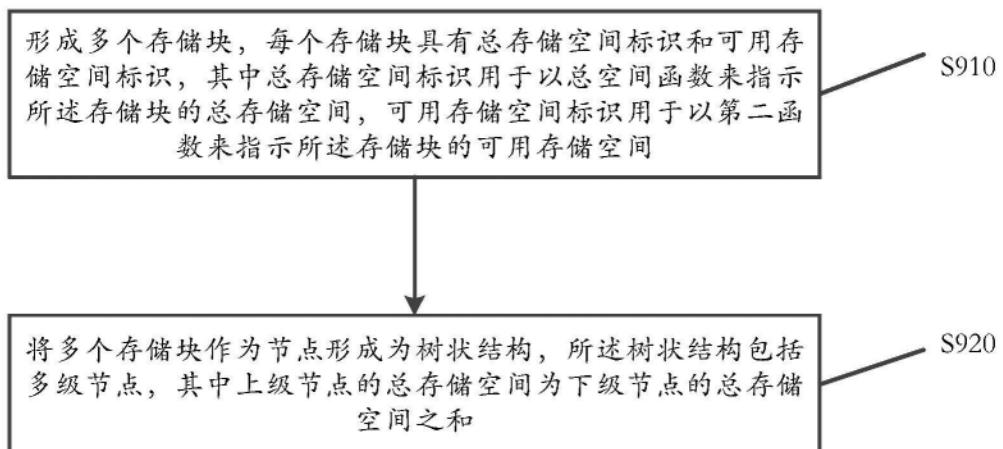


图9

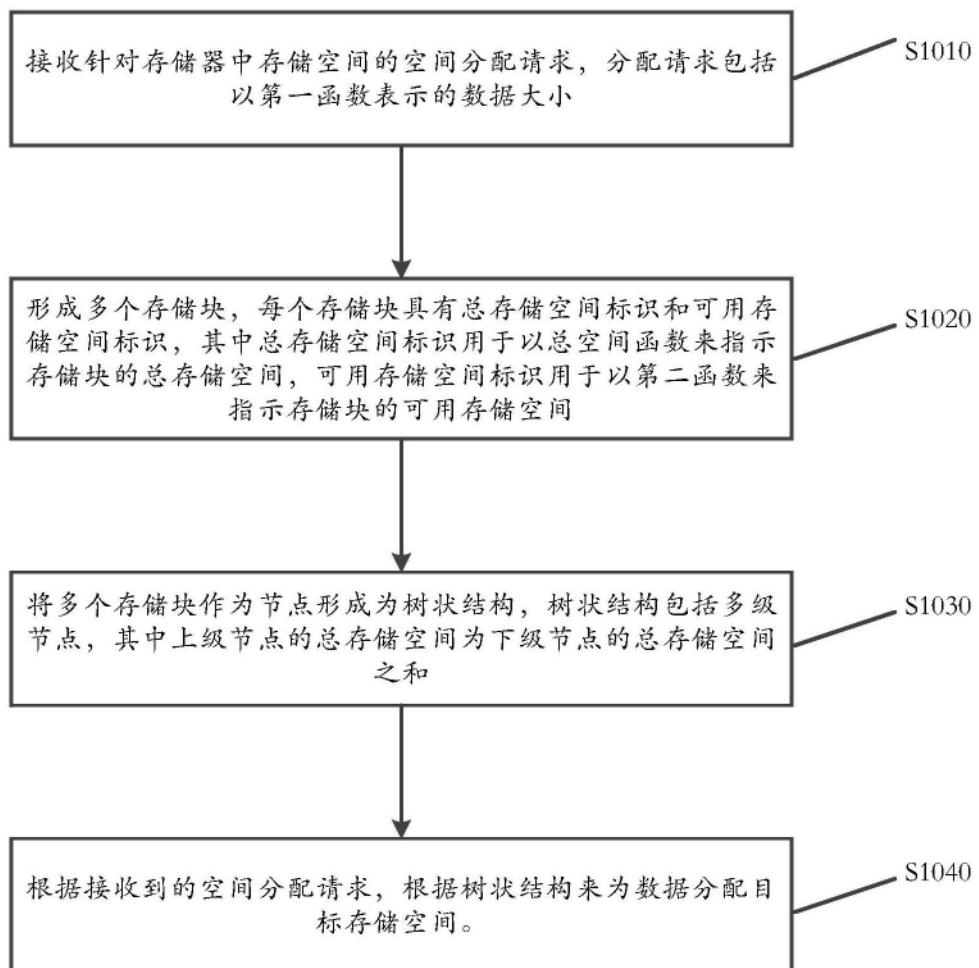


图10

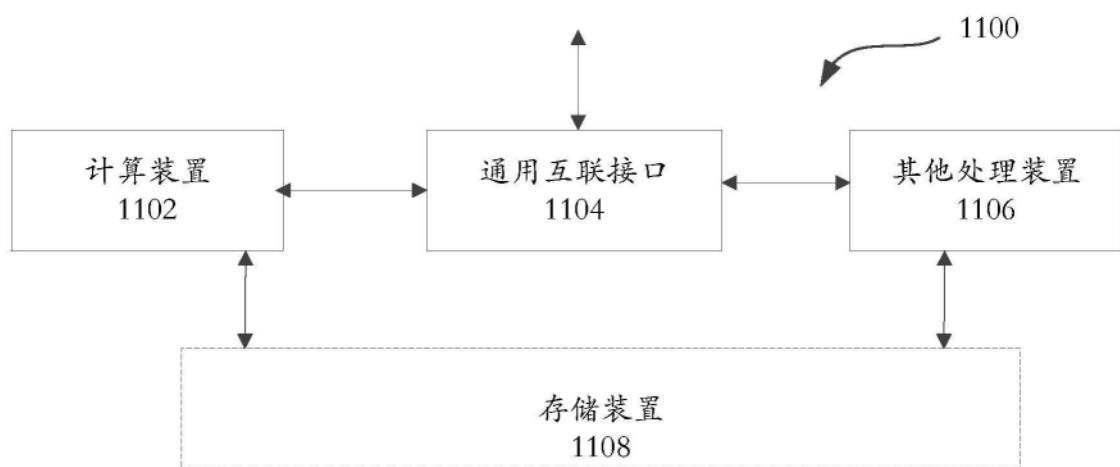


图11

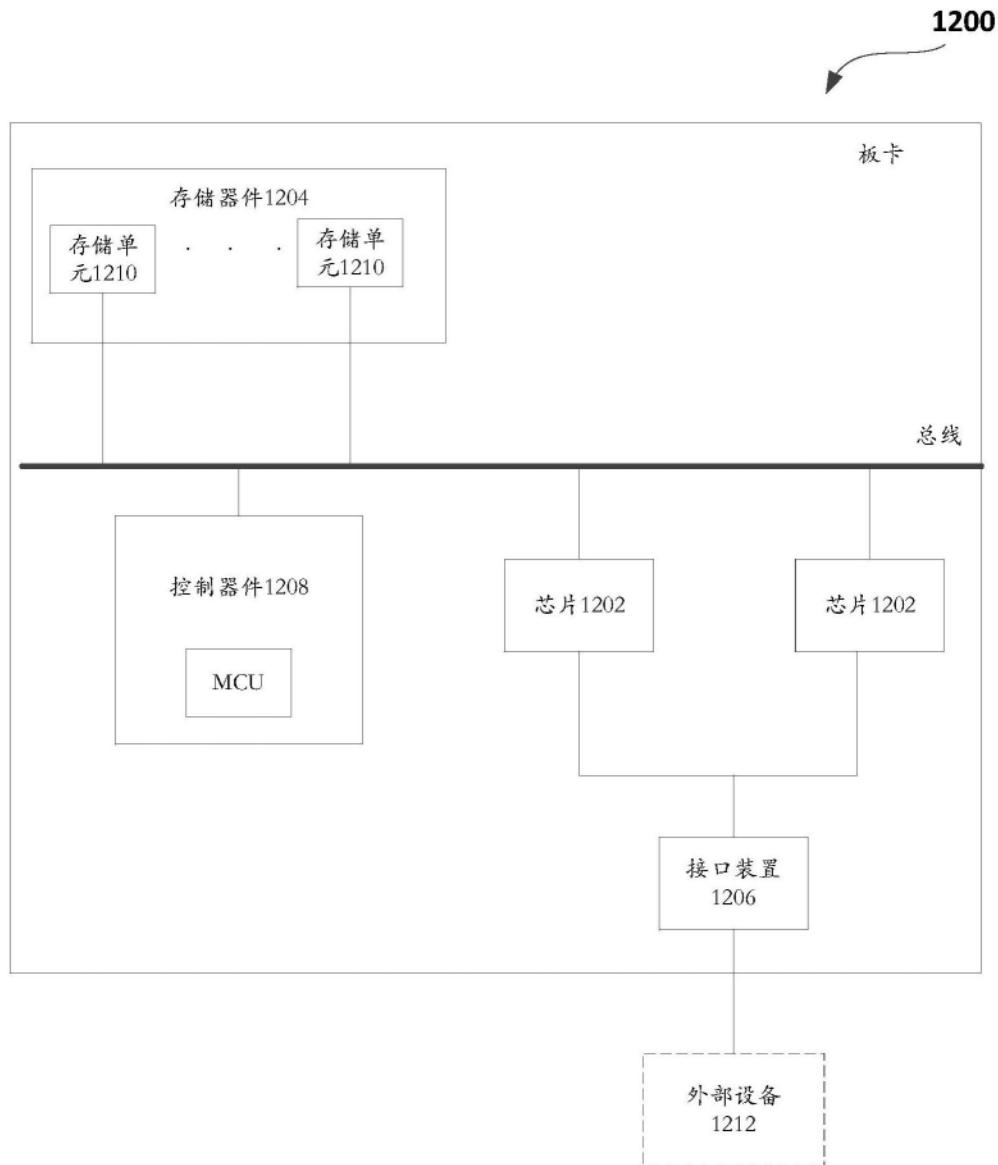


图12