(19)国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 121664997 A
(43)申请公布日 2026.03.13

(21)申请号 202511627028.2
(22)申请日 2025.11.07
(66)本国优先权数据
202411591541.6 2024.11.08 CN
(71)申请人 北京大学
地址 100871 北京市海淀区颐和园路5号北京大学
(72)发明人 杨仝 江子涵 刘子瑞
(74)专利代理机构 北京君尚知识产权代理有限公司 11200
专利代理师 邱晓锋
(51)Int.Cl.
H04N 19/30(2014.01)
H04N 19/42(2014.01)
H04N 19/142(2014.01)
H04N 19/154(2014.01)

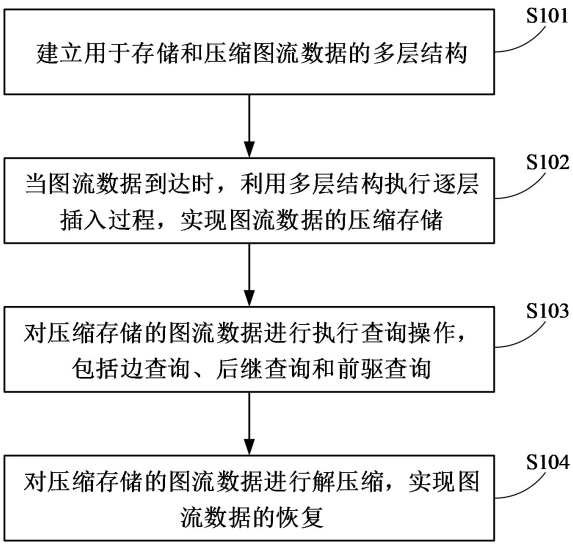
权利要求书2页 说明书6页 附图5页

(54)发明名称

一种基于压缩感知理论的图流数据压缩与恢复方法和系统

(57)摘要

本发明属于网络流量测量和网络数据处理领域,涉及一种基于压缩感知理论的图流数据压缩与恢复方法和系统。该方法包括:建立用于存储和压缩图流数据的多层结构;当图流数据到达时,利用多层结构执行逐层插入过程,实现图流数据的压缩存储;对压缩存储的图流数据进行执行查询操作,包括边查询、后继查询和前驱查询;对压缩存储的图流数据进行解压缩,实现图流数据的恢复。本发明采用分层架构设计,通过巧妙的编码方式和优化求解实现了对图流数据的高效压缩和准确恢复,本发明还支持常数复杂度下高效的前驱、后继权值和查询。



CN 121664997 A

1.一种基于压缩感知理论的图流数据压缩与恢复方法,其特征在于,包括以下步骤:

建立用于存储和压缩图流数据的多层结构;

当图流数据到达时,利用多层结构执行逐层插入过程,实现图流数据的压缩存储;

对压缩存储的图流数据进行执行查询操作,包括边查询、后继查询和前驱查询;

对压缩存储的图流数据进行解压缩,实现图流数据的恢复。

2.根据权利要求1所述的方法,其特征在于,所述多层结构的每一层由多个大小相同的块构成,每个块包含若干计数器,用于统计边的权值;每一层定义四个关键参数:block_size表示该层中每个块的大小,fp_len表示该层中计数器的位数大小,combine_ratio表示该层与下一层的合并比例,block_num表示该层中块的数量;第0层对应原始图流数据的邻接矩阵形式,其余层分为block_num块,每个块是一个(block_size, block_size)大小的矩阵。

3.根据权利要求1所述的方法,其特征在于,所述利用多层结构执行逐层插入过程,包括:

利用hash_num个哈希函数将图流数据映射到第i层的多个不同位置,对每个映射到的位置,将边的权值v加入对应的计数器中;

若某个计数器的值超过2^fp_len[i]-1,则发生溢出,其中fp_len[i]表示预先设置的第i层的计数器位数,此时保留低fp_len[i]位在当前层,并将溢出的高位部分递归插入到第i+1层,即使用哈希函数将溢出值映射到第i+1层的hash_num个位置,递归地再执行一次插入操作,每个位置插入的值为溢出值除以2^fp_len[i]得到的商。

4.根据权利要求1所述的方法,其特征在于,所述边查询包括:给定两个节点之间的一条边,在各层查找该边通过哈希函数映射的位置,同时递归查询上层以获取溢出值,最终将各层的查询结果组合得到边的实际权值。

5.根据权利要求1所述的方法,其特征在于,所述后继查询包括:扫描节点对应行的所有位置,为每个位置单独解压缩出对应的权值;将所有权值加和,即可得到后继查询的权值和结果;所述前驱查询包括:扫描节点对应列的所有位置,为每个位置单独解压缩出对应的权值;将所有权值加和,即可得到前驱查询的权值和结果。

6.根据权利要求1所述的方法,其特征在于,所述解压缩包括单点解压缩;所述单点解压缩包括:对第i层,查询第j块的(x, y)位置的权值;获取第i+1层中使用hash_num个哈希函数映射到的位置的值,对其取min值,得到当前(x, y)位置上的溢出值,再加上第i层的低位值,得到(x, y)位置上的原始值,通过递归查询到最顶层结束。

7.根据权利要求1所述的方法,其特征在于,所述解压缩包括全局解压缩;所述全局解压缩采用自顶向下的方式恢复完整的图结构,从最高层开始逐层向下进行恢复;对于第i层,通过第i+1层的计数器值和哈希映射关系构建线性方程组,其中方程组的变量为该第i层所有发生溢出位置的溢出值,通过求解L1范数最小化问题获得每个位置的精确溢出值,并递归执行直至第1层,最终准确恢复出原始的图结构。

8.一种基于压缩感知理论的图流数据压缩与恢复系统,其特征在于,包括:

多层结构构建模块,用于建立用于存储和压缩图流数据的多层结构;

插入模块,用于当图流数据到达时,利用多层结构执行逐层插入过程,实现图流数据的压缩存储;

查询模块,用于对压缩存储的图流数据进行执行查询操作,包括边查询、后继查询和前驱查询;

恢复模块,用于对压缩存储的图流数据进行解压缩,实现图流数据的恢复。

9.一种计算机设备,其特征在于,包括存储器和处理器,所述存储器存储计算机程序,所述计算机程序被配置为由所述处理器执行,所述计算机程序包括用于执行权利要求1~7中任一项所述方法的指令。

10.一种计算机可读存储介质,其特征在于,所述计算机可读存储介质存储计算机程序,所述计算机程序被计算机执行时,实现权利要求1~7中任一项所述的方法。

一种基于压缩感知理论的图流数据压缩与恢复方法和系统

技术领域

[0001] 本发明涉及网络流量测量和网络数据处理领域,具体涉及一种基于压缩感知理论的图流数据压缩与恢复方法和系统,适用于实现在有限资源约束下对高速图流数据的精确压缩和查询。

背景技术

[0002] 随着网络规模的不断扩大和业务复杂度的不断提升,图流数据的处理在网络测量、社交网络分析、云计算故障诊断等多个领域发挥着重要作用。图流数据是一个连续的数据序列,其中每条数据包含源节点、目标节点以及边权值等信息。这些数据共同构成一个动态变化的图结构。在实际应用场景中,图流数据具有如下特点:

高速性:在大型数据中心场景下,核心交换机的带宽已达到3.2Tbps甚至更高,这意味着每秒钟会产生数百万条边的更新。

[0003] 稀疏性:虽然图中可能的节点对数量庞大(例如对于二元组<源IP,目的IP>可达2^64种可能),但实际活跃的边数量相对较少,呈现出高度稀疏的特性。

[0004] 动态性:图的结构随着每条新数据的到来而持续变化,需要实时处理和更新。

[0005] 目前主流的图流处理方案主要包括两类:

第一类是基于采样的方案。这类方案通过预设的采样比对数据流进行采样,并将采样数据发送到控制面进行恢复和分析。然而,由于控制面接收数据包的性能限制(通常为每秒2百万个数据包),此类方案难以应对高速图流场景。

[0006] 第二类是基于Sketch(概要)的方案。这类方案在数据面维护紧凑的数据结构来记录图的信息。代表性工作包括CountMin Sketch、gSketch等。但现有Sketch方案存在以下不足:

精度有限:虽然这些方案提供了理论误差界限,但实际上只有少量流(如Heavy Hitter)能够达到较高精度,大量小流的统计精度很差。

[0007] 功能受限:部分Sketch方案(如CountMin、gSketch)无法支持图的拓扑相关查询,如可达性查询、后继查询等。

[0008] 资源开销大:为了保证精度,现有方案往往需要较大的存储空间,这与交换机有限的资源约束相矛盾。

[0009] 此外,已有的基于压缩感知理论的流量测量方案主要针对特定场景如流量矩阵或网络层析等,缺乏对通用图流数据的有效支持。因此,如何在有限资源约束下对高速图流数据进行精确压缩和查询,是一个亟待解决的关键问题。

发明内容

[0010] 为了解决在有限资源约束下对高速图流数据进行精确压缩和查询的问题,本发明提供一种基于压缩感知理论的图流数据压缩与恢复方法和系统。

[0011] 本发明的目的通过如下技术方案来实现:

一种基于压缩感知理论的图流数据压缩与恢复方法,包括以下步骤:

建立用于存储和压缩图流数据的多层结构;

当图流数据到达时,利用多层结构执行逐层插入过程,实现图流数据的压缩存储;

对压缩存储的图流数据进行执行查询操作,包括边查询、后继查询和前驱查询;

对压缩存储的图流数据进行解压缩,实现图流数据的恢复。

[0012] 进一步地,所述多层结构的每一层由多个大小相同的块构成,每个块包含若干计数器,用于统计边的权值;每一层定义四个关键参数:block_size表示该层中每个块的大小,fp_len表示该层中计数器的位数大小,combine_ratio表示该层与下一层的合并比例,block_num表示该层中块的数量;第0层对应原始图流数据的邻接矩阵形式,其余层分为block_num块,每个块是一个(block_size, block_size)大小的矩阵。

[0013] 进一步地,所述利用多层结构执行逐层插入过程,包括:

利用hash_num个哈希函数将图流数据映射到第i层的多个不同位置,对每个映射到的位置,将边的权值v加入对应的计数器中;

若某个计数器的值超过2^fp_len[i]-1,则发生溢出,其中fp_len[i]表示预先设置的第i层的计数器位数,此时保留低fp_len[i]位在当前层,并将溢出的高位部分递归插入到第i+1层,即使用哈希函数将溢出值映射到第i+1层的hash_num个位置,递归地再执行一次插入操作,每个位置插入的值为溢出值除以2^fp_len[i]得到的商。

[0014] 进一步地,所述边查询包括:给定两个节点之间的一条边,在各层查找该边通过哈希函数映射的位置,同时递归查询上层以获取溢出值,最终将各层的查询结果组合得到边的实际权值。

[0015] 进一步地,所述后继查询包括:扫描节点对应行的所有位置,为每个位置单独解压缩出对应的权值;将所有权值加和,即可得到后继查询的权值和结果;所述前驱查询包括:扫描节点对应列的所有位置,为每个位置单独解压缩出对应的权值;将所有权值加和,即可得到前驱查询的权值和结果。

[0016] 进一步地,所述解压缩包括单点解压缩;所述单点解压缩包括:对第i层,查询第j块的(x, y)位置的权值;获取第i+1层中使用hash_num个哈希函数映射到的位置的值,对其取min值,得到当前(x, y)位置上的溢出值,再加上第i层的低位值,得到(x, y)位置上的原始值,通过递归查询到最顶层结束。

[0017] 进一步地,所述解压缩包括全局解压缩;所述全局解压缩采用自顶向下的方式恢复完整的图结构,从最高层开始逐层向下进行恢复;对于第i层,通过第i+1层的计数器值和哈希映射关系构建线性方程组,其中方程组的变量为该第i层所有发生溢出位置的溢出值,通过求解L1范数最小化问题获得每个位置的精确溢出值,并递归执行直至第1层,最终准确恢复出原始的图结构。

[0018] 一种基于压缩感知理论的图流数据压缩与恢复系统,其包括:

多层结构构建模块,用于建立用于存储和压缩图流数据的多层结构;

插入模块,用于当图流数据到达时,利用多层结构执行逐层插入过程,实现图流数据的压缩存储;

查询模块,用于对压缩存储的图流数据进行执行查询操作,包括边查询、后继查询和前驱查询;

恢复模块,用于对压缩存储的图流数据进行解压缩,实现图流数据的恢复。

[0019] 本发明的有益效果是:

本发明采用分层架构设计,通过巧妙的编码方式和优化求解实现了对图流数据的高效压缩和准确恢复。在实验场景下,本发明架构所占内存空间仅为边数的约1.13倍,实现了空间复杂度平方级别到近线性级别的压缩;同时除单纯的图流压缩与恢复外,本发明架构还支持常数复杂度下高效的前驱、后继权值和查询。

附图说明

[0020] 图1是本发明的一种基于压缩感知理论的图流数据压缩与恢复方法的步骤流程图。

[0021] 图2是本发明的用于存储和压缩图流数据的多层结构示例图。

[0022] 图3是图流数据插入的一个示例图。

[0023] 图4是图流数据插入的另一个示例图。

[0024] 图5是图流数据插入后的结果示例图。

[0025] 图6是查询后继边权和的示例图。

[0026] 图7是本发明的一种基于压缩感知理论的图流数据压缩与恢复系统的模块构成图。

具体实施方式

[0027] 为使本发明的上述目的、特征和优点能够更加明显易懂,下面通过具体实施例和附图,对本发明做进一步详细说明。

[0028] 本发明提供一种基于压缩感知理论的图流数据压缩与恢复方法。其中压缩感知理论是指只要信号是“可压缩”的(或稀疏的),就可以用远低于传统奈奎斯特采样定理所要求的采样率来获取信号,并能从这些少量的采样数据中精确地重建出原始信号。

[0029] 本发明可以实现图流数据的实时插入和删除,插入和删除的结果均以压缩的形式呈现,可以在任意一个中间步骤解压缩,并且若收到查询请求时,可以实时得到查询节点的前驱、后继权值和。原始的、未压缩的图流数据以邻接矩阵的形式存在,这是一个行数与列数均为节点数的矩阵,(i, j)上的非零数值v表示节点i到节点j有一条权值为v的边。

[0030] 图1是本发明的一种基于压缩感知理论的图流数据压缩与恢复方法的步骤流程图,该方法包括以下步骤:

步骤S101:建立用于存储和压缩图流数据的多层结构。

[0031] 本发明采用一个多层结构来存储和压缩图流数据。每一层由多个大小相同的块构成,每个块包含若干计数器,用于统计边的权值。为了实现层与层之间的关联,为每一层定义四个关键参数:block_size表示该层中每个块的大小,fp_len表示该层中计数器的位数大小,combine_ratio表示该层与下一层的合并比例,block_num表示该层中块的数量。其中第0层对应原始图流数据的邻接矩阵形式,其余层分为block_num块,每个块是一个(block_size, block_size)大小的矩阵。因此,第0层也认为可是一个block_num为1,block_size为节点数的结构。图2为多层结构的示例,示意了三层结构。

[0032] 步骤S102:当图流数据到达时,利用多层结构执行逐层插入过程,实现图流数据的

压缩存储。

[0033] 当一条图流数据到达时,执行逐层插入过程。假设现在要向第i层插入一条边,边有对应的权值,这样一次插入可以用(x, y, v)来表示,即节点x到节点y有一条权值为v的边。首先利用hash_num个哈希函数将图流数据映射到第i层的多个不同位置。对每个映射到的位置,将边的权值也就是v加入对应的计数器中。若某个计数器的值超过了2^fp_len[i]-1,则发生溢出,其中fp_len[i]表示预先设置好的第i层的计数器位数。此时,保留低fp_len[i]位在当前层,并将溢出的高位部分递归插入到第i+1层。具体地,使用哈希函数将溢出值映射到第i+1层的hash_num个位置,递归地再执行一次插入操作,每个位置插入的值为溢出值除以2^fp_len[i]得到的商。这个过程一直持续到不再发生溢出或达到最高层为止。

[0034] 步骤S103:对压缩存储的图流数据进行执行查询操作,包括边查询、后继查询和前驱查询。

[0035] 对于查询操作,本发明支持三种基本的图操作:边查询、后继查询和前驱查询。在边查询中,给定节点s和节点d之间的一条边(s,d),在各层查找该边通过哈希函数映射的位置,同时需要递归查询上层以获取溢出值,最终将各层的查询结果组合得到边的实际权值。在后继查询中,给定一个节点v,需要计算其所有出边的权值之和。为此,在各层查找节点v对应行的所有可能位置,对每个位置递归处理可能的溢出情况,最后返回所有查询结果中的最小值作为最终结果。前驱查询的过程与后继查询类似,只是需要查找节点对应列的位置而非行的位置。

[0036] 具体地,后继查询包括以下步骤:

1)扫描节点v对应行的所有位置,为每个位置单独解压缩出对应的权值;

2)将所有权值加和,即可得到后继查询的权值和结果。

[0037] 具体地,前驱查询包括以下步骤:

1)扫描节点v对应列的所有位置,为每个位置单独解压缩出对应的权值;

2)将所有权值加和,即可得到前驱查询的权值和结果。

[0038] 步骤S104:对压缩存储的图流数据进行解压缩,实现图流数据的恢复。

[0039] 上述两种查询都用到了解压缩这一核心操作,解压缩又分为单点解压缩与全局解压缩。

[0040] 对于单点解压缩,分层递归地对多层结构进行查询。具体而言,单点解压缩可以拆分为多个独立查询过程,具体定义为:对第i层,查询第j块的(x, y)位置的权值。对于这个操作,由于在插入时,若(x, y)位置发生溢出,会将溢出结果插入到第i+1层中,因此需要得到第i+1层中使用hash_num个哈希函数映射到的位置的值,对其取min值,即可得到当前(x, y)位置上的溢出值,再加上第i层的低位值,即可得到(x, y)位置上的原始值。这是一个递归的过程,对原始邻接矩阵的查询实际上可以看做是对第0层的一次查询,随后这次查询会在第1层上计算结果,第1层又会触发第2层的查询,查询到最顶层结束。

[0041] 对于全局解压缩,为了恢复完整的图结构,本发明采用一种自顶向下的恢复方法。首先从最高层开始,逐层向下进行恢复。对于第i层,通过第i+1层的计数器值和哈希映射关系构建线性方程组,其中方程组的变量为该第i层所有发生溢出位置的溢出值。这个问题可以转化为一个L1范数最小化问题,通过优化求解来获得每个位置的精确溢出值。通过递归执行这个过程直至第1层,最终可以准确恢复出原始的图结构。

[0042] 此外,对于所有的查询过程,还可以使用Bitmap(位图)或BloomFilter(布隆过滤器)来标识是否溢出,实现更精准的方程组构造和更快的查询。

[0043] 下面提供本发明的一个实施例。本实施例中,定义一个4个点构成的图流,利用本发明中提出的多层结构压缩图流,假设layer 1(第一层)的block_num为1,block_size为4,fp_len为2,combine_ratio为1,layer 2(第二层)的block_num为1,block_size为2,fp_len为2,总体hash_num为2。如图2所示。

[0044] 1)现在插入(1, 2, 3),即节点1到节点2有一条权值为3的边。

[0045] 如图3所示,layer 0是真实的邻接矩阵,需要将其利用哈希函数映射到layer 1的两个位置,假设选中的位置为图3中箭头指向的位置,由于3并没有超过fp_len为2能表示的最大范围,因此此时没有发生溢出,插入结束。

[0046] 2)现在插入(2, 3, 2),即节点2到节点3有一条权值为2的边。

[0047] 真实的邻接矩阵需要加上2,利用哈希函数选到layer 1的两个地方,假设选中的地方为图4中箭头指向的地方,其中一个地方正常加2,红色数字是两次加起来3+2=5,fp_len为2只能表示到3,因此发生溢出,需要进一步将溢出值插入到layer 2,5可以表示为1 + (1<<2),所以layer 1中留下1,layer 2中需要利用哈希函数选两个位置插入1。最终图像结果如图5所示。

[0048] 3)考虑查询过程,由于前驱查询和后继查询基本为互逆操作,因此本实施例只考虑如何查询后继边权和。

[0049] 现在查询1的后继边权和。对应到邻接矩阵上,即为1对应行的所有值的和,由于是举例,只考虑有值的3是如何得到的,首先在layer 1中进行查询,查到虚线标注的两个位置上,两个位置又需要对layer 2进行递归查询,这里不展示数值3对应的虚线如何查询,对于5(在layer 1中存储的是1),需要在layer 2中查询两个位置,算出溢出的结果4(layer 2中的1表示layer 1中的4),和layer 1中存储的值相加即可得到5。3和5取min值(最小值)即可得到正确的后继边权和3。如图6所示。

[0050] 4)考虑如何恢复整个邻接矩阵。

[0051] 通过逆向一层一层恢复的方法逐层恢复,layer 2无需恢复,考虑如何利用layer 2恢复layer 1。layer 2中的每个数字,都对应layer 1的一些格子,假设这些layer 1的格子是(x1, y1) (x2, y2) … (xn, yn),溢出值用overflow表示,layer 2的数字是Z,则overflow (x1, y1) + overflow (x2,y2) + … + overflow (xn, yn) = Z,得到所有方程之后,就可以使用解优化问题的方法求出layer 1中的所有溢出值。

[0052] 在这个例子中,其中一个方程为overflow(2, 2) + overflow(…) = 4,(2, 2)是明确已知发生溢出的部分,因此这里用于举例说明,overflow(…)表示同样通过哈希函数映射到layer 2相同格子的其他layer 1的溢出值。

[0053] 本发明的图流数据压缩与恢复方法,可应用于网络测量、社交网络分析、云计算故障诊断等领域。

[0054] 例如,本发明的一个实施例提供一种网络测量方法,用于测量大规模数据中心或骨干网中的实时流量矩阵。具体实施时,采用本发明方法对高速网络中以<源IP, 目的IP>对为边、流量大小为权值构成的图流数据,在交换机数据面进行实时压缩存储,进而在控制面通过解压缩恢复完整的流量矩阵,或直接对压缩结构进行“后继查询”以获取任一IP的总

出口流量,或进行“前驱查询”获取总入口流量,从而得到最终的网络测量结果。在交换机有限SRAM资源约束下,采用本发明方法能够实现全网流量的近线性空间开销和高精度实时监控。

[0055] 例如,本发明的一个实施例提供一种社交网络分析方法,用于分析大型社交平台(如Twitter, 微博)上的实时话题传播和用户影响力。具体实施时,采用本发明方法对社交网络中的用户间的关注、转发、评论等行为构成的动态图流数据进行高效压缩和分布式存储,进而通过“后继查询”实时计算某一用户的潜在传播范围(出度权值和),或通过“前驱查询”追溯某一热点话题的影响力来源(入度权值和),最终得到实时的用户影响力排名和信息传播路径图。可以解决现有技术中全量存储海量社交图数据导致成本高昂,或传统Sketch方案无法支持复杂拓扑查询的问题,能够低成本、高精度地实时洞察网络动态和关键节点。

[0056] 本发明的另一个实施例提供一种基于压缩感知理论的图流数据压缩与恢复系统,如图7所示,其包括:

多层结构构建模块201,用于建立用于存储和压缩图流数据的多层结构;

插入模块202,用于当图流数据到达时,利用多层结构执行逐层插入过程,实现图流数据的压缩存储;

查询模块203,用于对压缩存储的图流数据进行执行查询操作,包括边查询、后继查询和前驱查询;

恢复模块204,用于对压缩存储的图流数据进行解压缩,实现图流数据的恢复。

[0057] 上述各模块的划分仅是举例说明,实际应用中可以根据需要而将上述功能分配由不同的功能模块完成,以完成前述方法中描述的全部或者部分功能。上述各模块的具体工作过程,可以参考前述方法实施例中的对应过程,在此不再赘述。

[0058] 应该理解到,在本发明所提供的上述实施例中所揭露的方法和系统,可以通过其它的方式实现。例如,以上模块的划分在实际实现时可以有另外的划分方式,多个模块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。本发明中的各个步骤、各个模块可以软件功能单元的形式实现,可以存储在一个计算机可读取存储介质中,包括若干指令用以使得一台计算机设备执行本发明所述方法的部分或全部步骤。例如本发明的一实施例提供一种计算机设备(计算机、服务器、智能手机等),其包括存储器和处理器,所述存储器存储计算机程序,所述计算机程序被配置为由所述处理器执行,所述计算机程序包括用于执行本发明方法中各步骤的指令。例如本发明的另一实施例提供一种计算机可读存储介质(如ROM/RAM、磁盘、光盘等),所述计算机可读存储介质存储计算机程序,所述计算机程序被计算机执行时,实现本发明方法的各个步骤。例如本发明的另一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计算机程序,所述计算机程序被计算机执行时,实现本发明方法的步骤。

[0059] 以上公开的本发明的具体实施例,其目的在于帮助理解本发明的内容并据以实施,本领域的普通技术人员可以理解,在不脱离本发明的精神和范围内,各种替换、变化和修改都是可能的。本发明不应局限于本说明书的实施例所公开的内容,本发明的保护范围以权利要求书界定的范围为准。

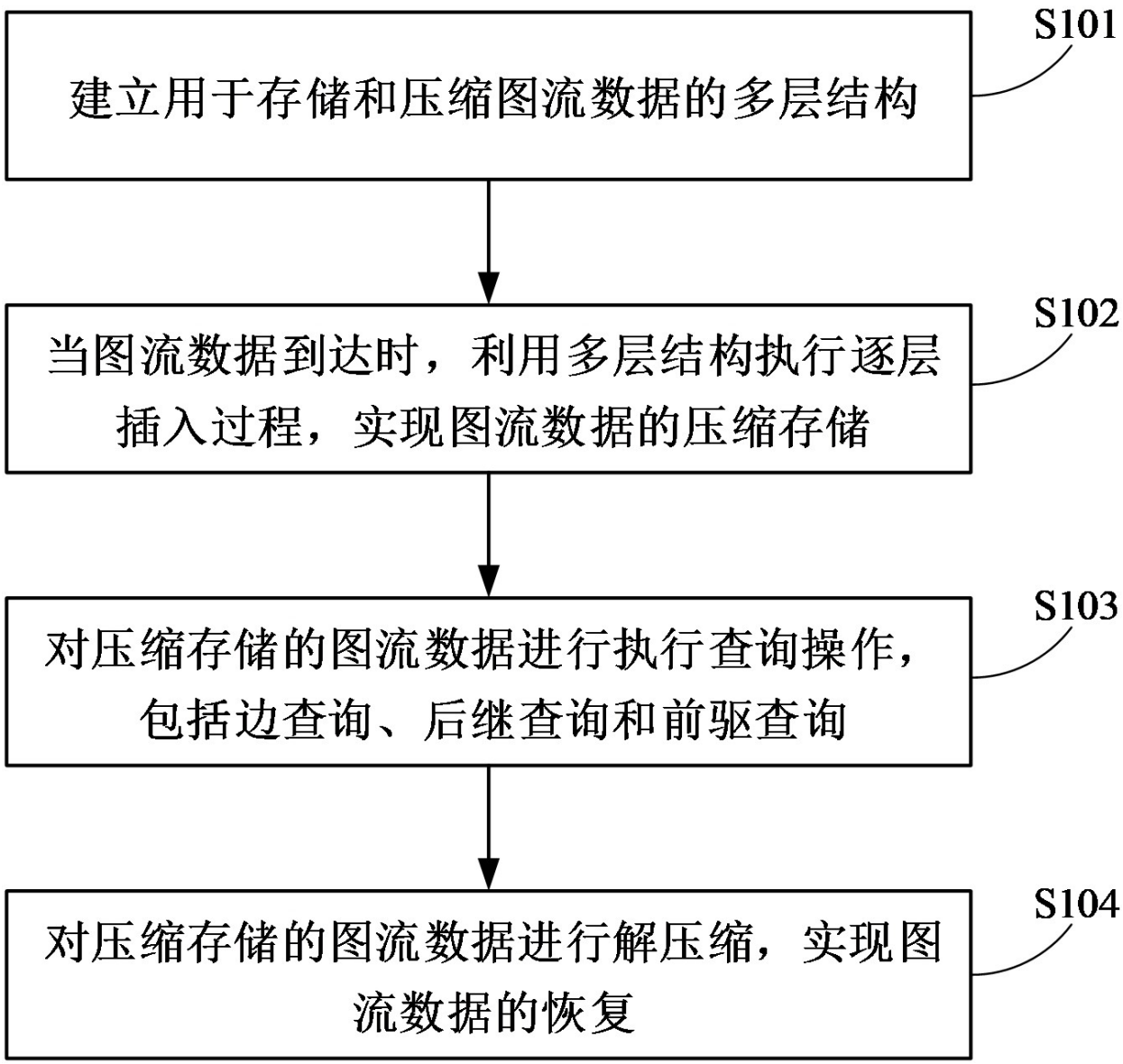

S101
建立用于存储和压缩图流数据的多层结构
S102
当图流数据到达时，利用多层结构执行逐层插入过程，实现图流数据的压缩存储
S103
对压缩存储的图流数据进行执行查询操作，包括边查询、后继查询和前驱查询
S104
对压缩存储的图流数据进行解压缩，实现图流数据的恢复

图1

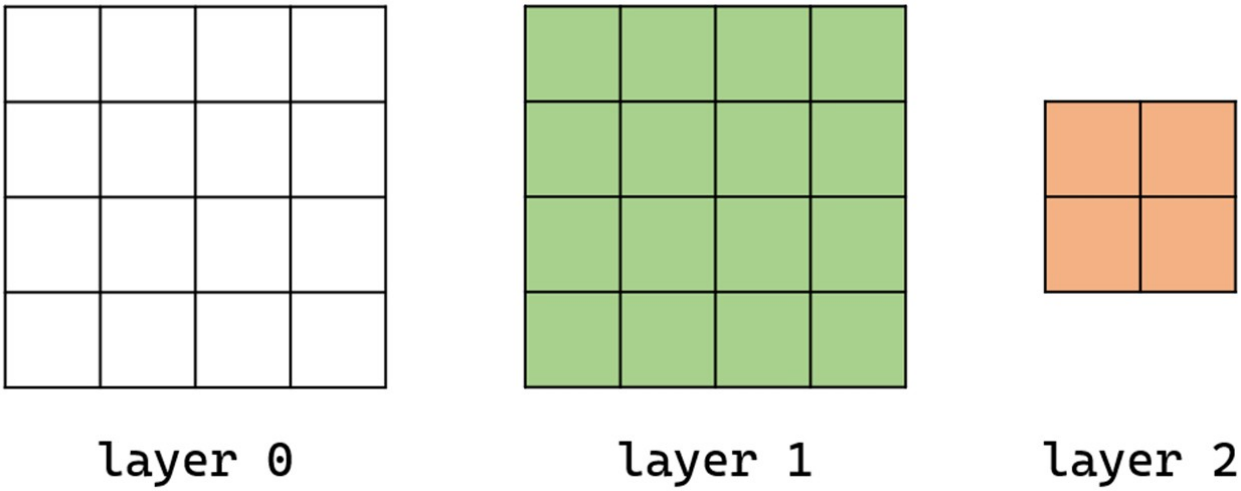


图2

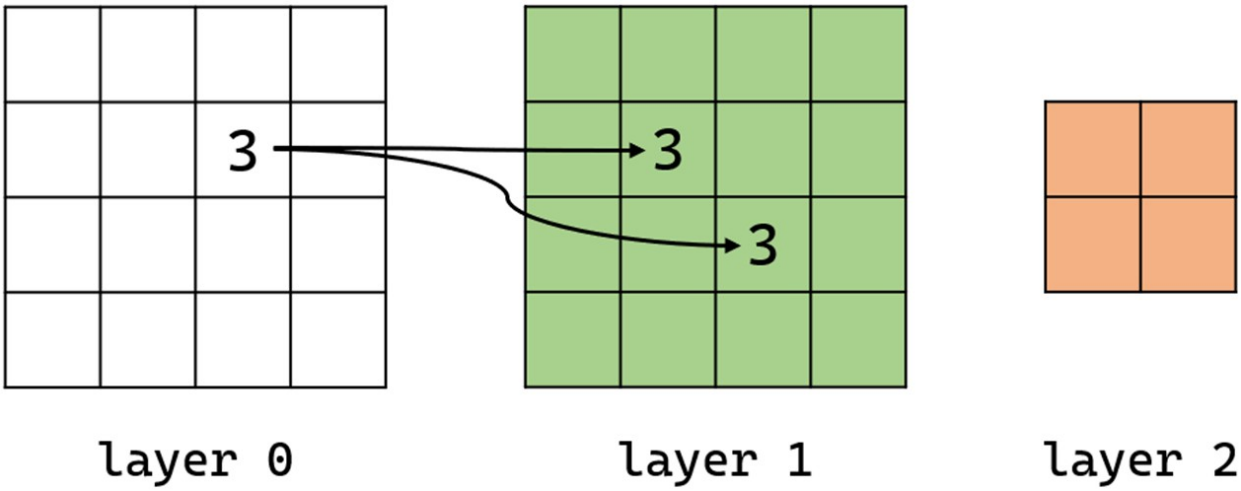


图3

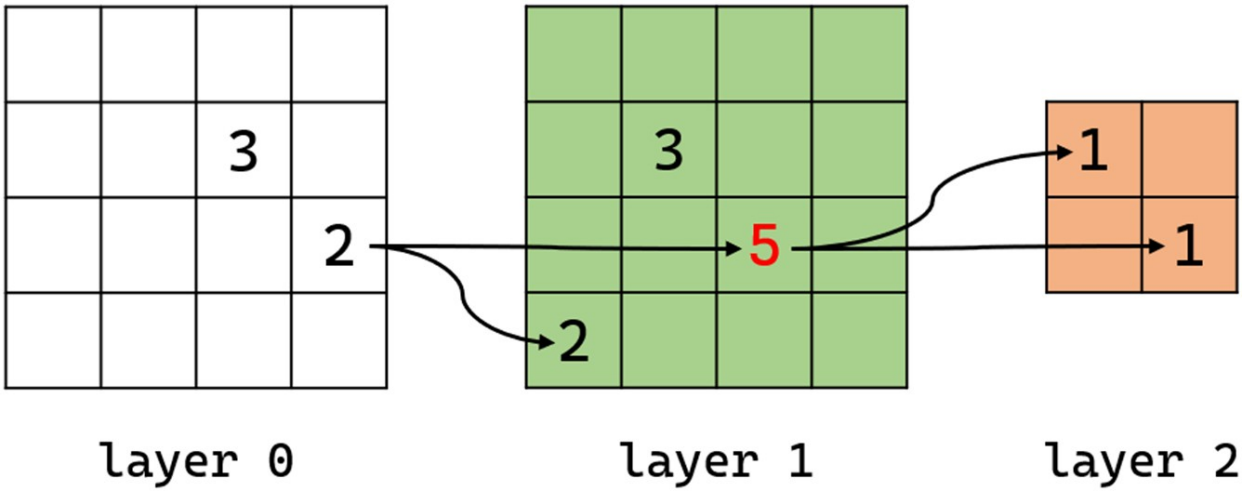

3
2
3
5
2
1
1
layer 0
layer 1
layer 2

图4

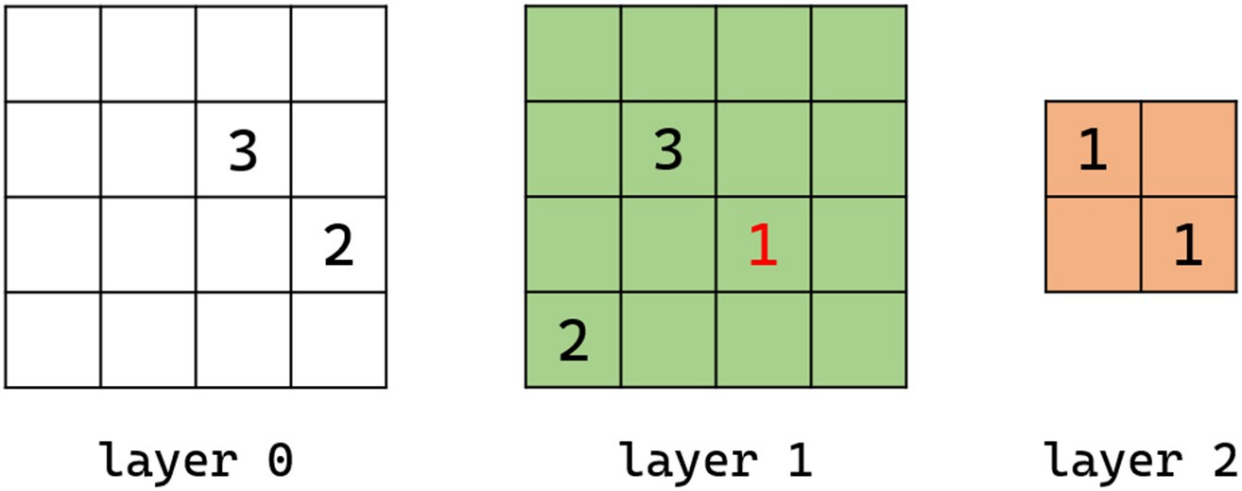

3
2
3
1
2
1
1
layer 0
layer 1
layer 2

图5

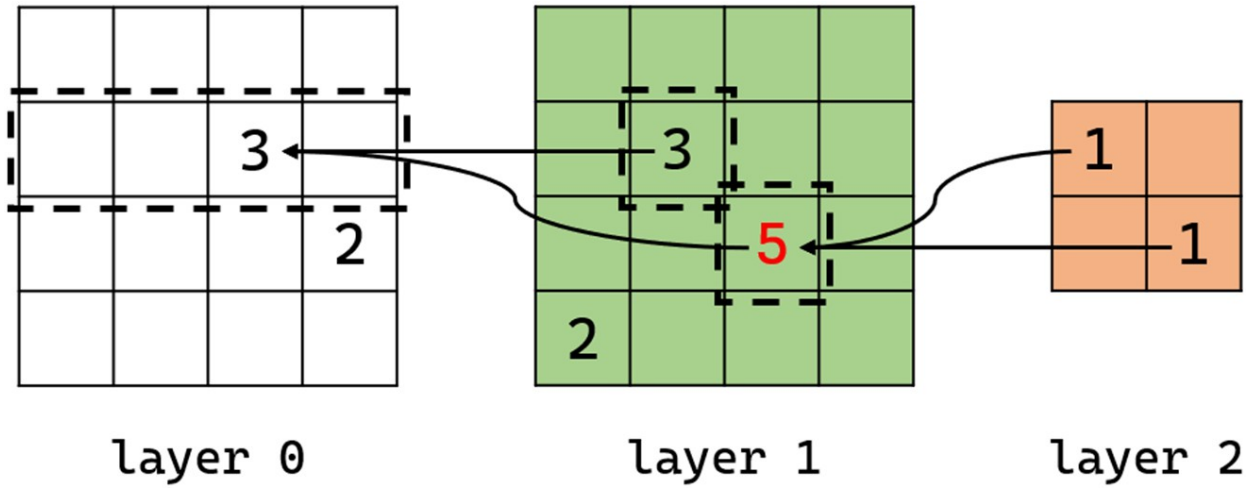


图6

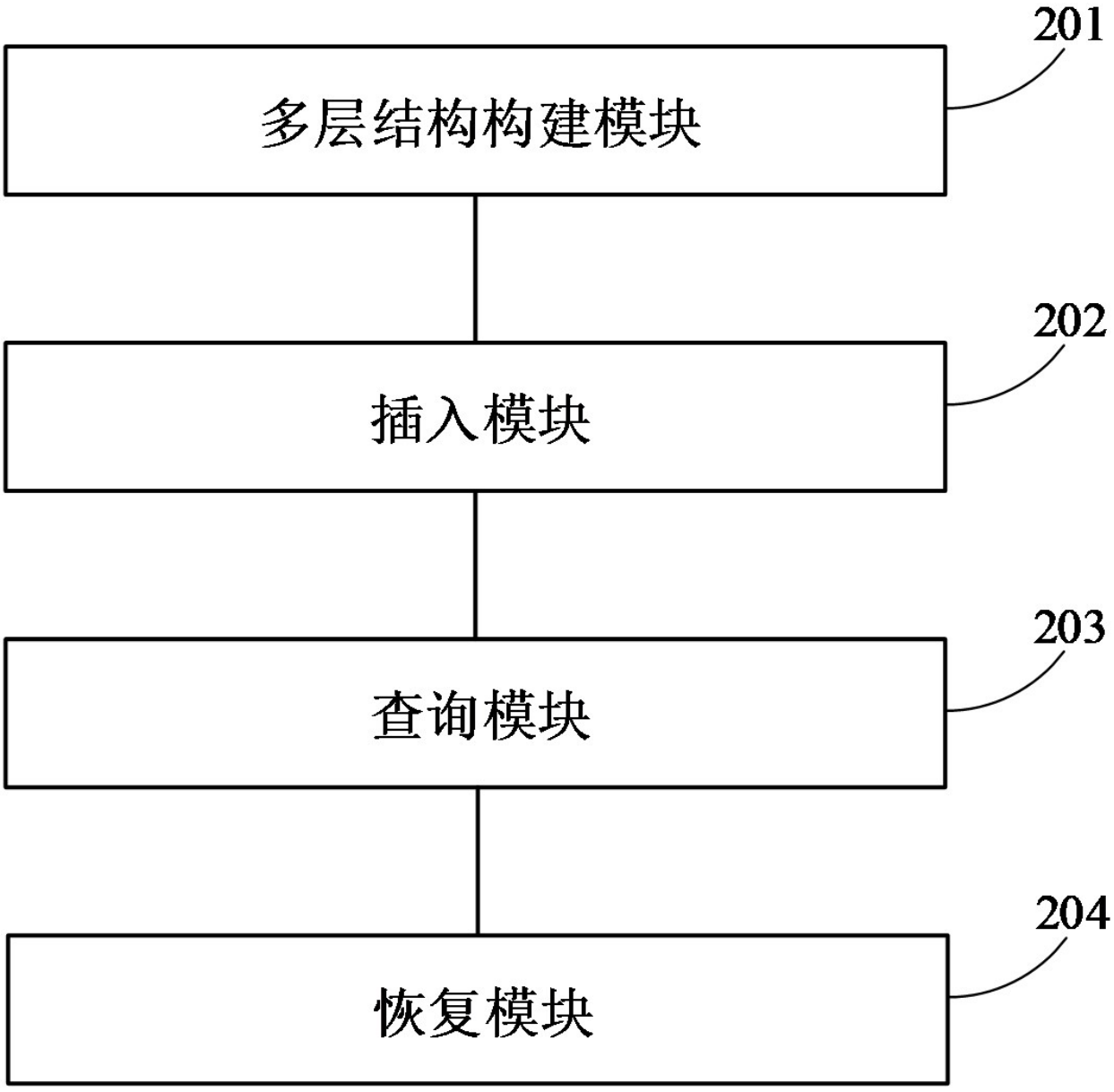


图7