



(12)发明专利

(10)授权公告号 CN 105095371 B

(45)授权公告日 2018.08.10

(21)申请号 201510370342.7

审查员 王秋平

(22)申请日 2015.06.29

(65)同一申请的已公布的文献号

申请公布号 CN 105095371 A

(43)申请公布日 2015.11.25

(73)专利权人 清华大学

地址 100084 北京市海淀区清华大学

专利权人 中国科学院计算技术研究所

(72)发明人 韩文弢 李恺威 陈世敏 陈文光

郑纬民

(74)专利代理机构 北京睿邦知识产权代理事务

所(普通合伙) 11481

代理人 张丽新

(51)Int.Cl.

G06F 17/30(2006.01)

权利要求书4页 说明书14页 附图7页

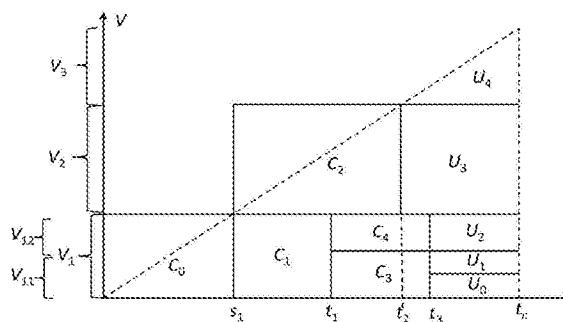
(54)发明名称

时序图的图数据管理方法及其装置

(57)摘要

提供了时序图的图数据管理方法和装置,图由顶点和边组成。图数据管理方法包括:获得时序图的事件数据;以及以二维空间-时间数据块C=(Vc,Tc)形式组织时序图的数据并存储在存储设备上,一个维度是时间维度,另一个维度是顶点维度,数据块C=(Vc,Tc)保存一个时间区间[s\_c,t\_c]中与顶点集合Vc相关的数据,所述数据块C=(Vc,Tc)逻辑上包括与顶点集合Vc相关的、在时刻s\_c处图的快照以及在时间区间[s\_c,t\_c]内发生事件的日志,其中Vc是顶点集合,Tc指示时间区间,Tc=[s\_c,t\_c],s\_c表示该时间区间的起始时刻,t\_c表示该时间区间的结束时刻。根据本发明实施例的时序图的图数据管理方法和图数据管理装置,以二维空间-时间数据块形式组织时序图,适合于高效地存储和查询时序图数据。

4000



1. 一种时序图的图数据管理方法,图由顶点和边组成,包括:

获得时序图的事件数据;以及

以二维空间-时间数据块 $C = (V_c, T_c)$ 形式组织时序图数据并存储在存储设备上,一个维度是时间维度,另一个维度是顶点维度,数据块 $C = (V_c, T_c)$ 保存一个时间区间 $[s_c, t_c]$ 中与顶点集合 $V_c$ 相关的数据,所述数据块 $C = (V_c, T_c)$ 逻辑上包括与顶点集合 $V_c$ 相关的在时刻 $s_c$ 处图的快照以及在时间区间 $[s_c, t_c]$ 内发生事件的日志,其中 $V_c$ 是顶点集合, $T_c$ 指示时间区间, $T_c = [s_c, t_c]$ , $s_c$ 表示该时间区间的起始时刻, $t_c$ 表示该时间区间的结束时刻,

其中在顶点维度上,针对已经存在的每个顶点集合,时间上从该顶点集合的最后一个空间-时间数据块 $C_n$ 的结束时刻 $t_{ne}$ 开始,存储结束时刻 $t_{ne}$ 的快照作为与该顶点集合相关联的最新快照,以及随时间进行,接收新发生的事件,并以日志形式存储,

关于该顶点集合,以最后一个空间-时间数据块 $C_n$ 的结束时刻 $t_{ne}$ 时刻作为下一空间-时间数据块 $C_{n+1}$ 的开始时刻 $t_{(n+1)s}$ ,如下确定下一空间-时间数据块 $C_{(n+1)}$ 的结束时刻 $t_{(n+1)e}$ :

设 $S_u$ 为下一空间-时间数据块 $C_{(n+1)}$ 的开始时刻 $t_{(n+1)s}$ 的快照的大小, $L_u$ 为从所述开始时刻 $t_{(n+1)s}$ 起到目前为止、关于最后一个空间-时间数据块 $C_n$ 的顶点集合存储的日志的大小,当 $L_u/S_u \geq \lambda$ 且 $L_u \geq \gamma$ 时,基于开始时刻 $t_{(n+1)s}$ 的快照和所述从所述开始时刻 $t_{(n+1)s}$ 起到目前为止、关于最后一个空间-时间数据块 $C_n$ 的顶点集合存储的日志,形成下一空间-时间数据块 $C_{(n+1)}$ ,其中 $\lambda$ 为类底数参数, $\gamma$ 为最小的分割阈值。

2. 根据权利要求1的图数据管理方法,与顶点集合 $V_c$ 相关的、在时刻 $s_c$ 处图的快照逻辑上以下列两种形式之一表示:

在时刻 $s_c$ 处点和边的连接情况和属性设置情况;

在时刻 $s_c$ 处有效的事件的数据的集合。

3. 根据权利要求1的图数据管理方法,还包括:

维持全局数据块索引,基于该全局数据块索引,给定顶点ID和时刻,能够定位到与该顶点ID和该时刻相关的数据在哪个数据块。

4. 根据权利要求3的图数据管理方法,所述空间-时间数据块还包括数据块内部索引,在基于给定顶点ID和时刻、基于全局数据块索引定位到具体空间-时间数据块后,基于数据块内部索引,基于该给定顶点ID,能够定位到在该具体空间-时间数据块中与该给定顶点ID相关的具体数据段。

5. 根据权利要求4的图数据管理方法,所述空间-时间数据块 $C = (V_c, T_c)$ 物理上包括多个顶点的数据段,每个顶点的数据段包含该顶点的事件以及所有以该顶点作为端点的边相关的事件,所述顶点的事件包括关于该顶点的在时刻 $s_c$ 处有效的事件和在时间区间 $[s_c, t_c]$ 内发生的所有事件,以该顶点作为端点的边相关的事件包括关于该边的在时刻 $s_c$ 处有效的事件和在时间区间 $[s_c, t_c]$ 内发生的所有事件,每个事件信息包括事件内容和事件发生的时刻。

6. 根据权利要求5的图数据管理方法,所述多个顶点的数据段在物理上是一个顶点的数据段一个顶点的数据段地顺序存储的,每个顶点的数据段中的所有数据集中存储在一起。

7. 根据权利要求1的图数据管理方法,形成下一空间-时间数据块 $C_{(n+1)}$ 时,判断下一

空间-时间数据块 $C(n+1)$ 的结束时刻 $t(n+1)e$ 的快照的大小是否超过预定阈值,当超过预定阈值时,将空间-时间数据块 $C(n+1)$ 的顶点集合分割为两个不相交的第一顶点集合和第二顶点集合,后续将关于第一顶点集合和第二顶点集合分别形成对应的快照和日志,进而形成各自的空间-时间数据块。

8. 根据权利要求1的图数据管理方法,还包括:在形成空间-时间数据块时或空间-时间数据块已形成后,重新确定各个顶点的数据段进行物理存储的顺序,并按照所确定的顺序存储各个顶点的数据段。

9. 根据权利要求1的图数据管理方法,还包括:

对于新出现的不属于任何已经存在的顶点集合的顶点,维持一最新的顶点集合,最新的顶点集合中的顶点不属于任何空间-时间数据块相关联的顶点集合,将随时间进行新出现的顶点添加入该最新的顶点集合,并以日志形式存储关于该最新的顶点集合中各顶点的事件数据,当该日志数据大小超过预定阈值时,基于此时的最新的顶点集合和相关联的时间区间,形成空间-时间数据块;以及

将最新顶点集合清零,并接收后续出现的新顶点和相关联的事件数据。

10. 根据权利要求3的图数据管理方法,还包括:

接收查询,该查询涉及给定顶点或给定顶点集合和涉及给定时刻或给定时间区间;

基于该给定顶点或给定顶点集合和该给定时刻或给定时间区间,查询全局数据块索引,定位到相关联的空间-时间数据块;

对于定位到的相关联的空间-时间数据块的每个,

查询与该空间-时间数据块相关联的数据块内部索引,定位到与给定顶点或给定顶点集合相关联的具体数据段;以及

扫描该具体数据段,返回该具体数据段的查询结果;以及

合并各个查询结果,并返回合并得到的查询结果。

11. 一种图数据管理装置,包括:

事件摄入部件,配置为摄入新发生的事件,并发送至时序图数据管理引擎;

查询引擎,配置为接收来自外部的查询,并将该查询发送至时序图数据管理引擎,接收来自时序图数据管理引擎的查询结果,并输出该查询结果;以及

时序图数据管理引擎,以二维空间-时间数据块 $C=(V_c, T_c)$ 形式组织时序图的数据并存储在存储设备上,一个维度是时间维度,另一个维度是顶点维度,数据块 $C=(V_c, T_c)$ 保存一个时间区间 $[s_c, t_c]$ 中与顶点集合 $V_c$ 相关的数据,所述数据块 $C=(V_c, T_c)$ 逻辑上包括在时刻 $s_c$ 处图的快照以及在时间区间 $[s_c, t_c]$ 内发生事件的日志,其中 $V_c$ 是顶点集合, $T_c$ 指示时间区间, $T_c=[s_c, t_c]$ , $s_c$ 表示该时间区间的起始时刻, $t_c$ 表示该时间区间的结束时刻,在时刻 $s_c$ 处图的快照包括在时刻 $s_c$ 处有效的事件的数据的集合,

其中在顶点维度上,针对已经存在的每个顶点集合,时间上从该顶点集合的最后一个空间-时间数据块 $C_n$ 的结束时刻 $t_{ne}$ 开始,存储结束时刻 $t_{ne}$ 的快照作为与该顶点集合相关联的最新快照,以及随时间进行,接收新发生的事件,并以日志形式存储,

关于该顶点集合,以最后一个空间-时间数据块 $C_n$ 的结束时刻 $t_{ne}$ 时刻作为下一空间-时间数据块 $C_{n+1}$ 的开始时刻 $t(n+1)s$ ,如下确定下一空间-时间数据块 $C(n+1)$ 的结束时刻 $t(n+1)e$ :

设 $S_u$ 为下一空间-时间数据块 $C(n+1)$ 的开始时刻 $t(n+1)$ 的快照的大小, $L_u$ 为从所述开始时刻 $t(n+1)$ 起到目前为止、关于最后一个空间-时间数据块 $C_n$ 的顶点集合存储的日志的大小,当 $L_u/S_u \geq \lambda$ 且 $L_u \geq \gamma$ 时,基于开始时刻 $t(n+1)$ 的快照和所述从所述开始时刻 $t(n+1)$ 起到目前为止、关于最后一个空间-时间数据块 $C_n$ 的顶点集合存储的日志,形成下一空间-时间数据块 $C(n+1)$ ,其中 $\lambda$ 为类底数参数, $\gamma$ 为最小的分割阈值。

12. 一种时序图的图数据管理方法,图由顶点和边组成,该图数据管理方法包括:

以二维空间-时间数据块 $C = (V_c, T_c)$ 形式组织时序图的图数据并存储在存储设备上,一个维度是时间维度,另一个维度是顶点维度,数据块 $C = (V_c, T_c)$ 保存一个时间区间 $[s_c, t_c]$ 中与顶点集合 $V_c$ 相关的数据,所述数据块 $C = (V_c, T_c)$ 逻辑上包括在时刻 $s_c$ 处图的快照以及在时间区间 $[s_c, t_c]$ 内发生事件的日志,其中 $V_c$ 是顶点集合, $T_c$ 指示时间区间, $T_c = [s_c, t_c]$ , $s_c$ 表示该时间区间的起始时刻, $t_c$ 表示该时间区间的结束时刻,在时刻 $s_c$ 处图的快照包括在时刻 $s_c$ 处有效的事件的数据的集合;

将空间-时间数据块分类为封印的空间-时间数据块和未封印的空间-时间数据块,对于基于同样的顶点集合的封印的空间-时间数据块和未封印的空间-时间数据块,未封印的空间-时间数据块时间上晚于封印的空间-时间数据块,且保持开放状态以等待接收关于该顶点集合中的顶点的任何新发生的事件,而封印的空间-时间数据块已封闭不再接收新的事件;

对于任一未封印的空间-时间数据块,当满足预定的分割条件时,以满足预定的分割条件的时刻作为分界点,将该未封印的空间-时间数据块转变为新的封印的空间-时间数据块,该新的封印的空间-时间数据块以该分界点作为结束时刻,创建新的未封印的空间-时间数据块,该新的未封印的空间-时间数据块以该分界点为开始时刻,该新的封印的空间-时间数据块与该新的未封印的空间-时间数据块具有相同的顶点集合;以及

将未封印的空间-时间数据块分类为普通的未封印的空间-时间数据块和特殊的未封印的空间-时间数据块,普通的未封印的空间-时间数据块的顶点集合出现于封印的空间-时间数据块的顶点集合,特殊的未封印的空间-时间数据块的顶点集合中的顶点不曾出现于任何封印的空间-时间数据块,在任何当前时刻维持一个特殊的未封印的空间-时间数据块,当出现的事件涉及新顶点时,将该事件添加入该特殊的未封印的空间-时间数据块,当出现的事件涉及存在于普通的未封印的空间-时间数据块的顶点集合中的顶点时,将该事件添加入该普通的未封印的空间-时间数据块。

13. 根据权利要求12的图数据管理方法,还包括:

维持全局数据块索引和数据块内部索引,基于全局数据块索引,给定顶点ID和时刻,能够定位到具体空间-时间数据块,基于数据块内部索引,能够定位到在该具体空间-时间数据块中与该顶点相关的具体数据段。

14. 根据权利要求12的图数据管理方法,所述空间-时间数据块 $C = (V_c, T_c)$ 物理上包括顺序存储多个顶点的数据段和包括数据块内部索引,每个顶点的数据段包含该顶点的事件以及所有该顶点的边相关的事件,所述顶点的事件包括关于该顶点的在时刻 $s_c$ 处有效的事件和在时间区间 $[s_c, t_c]$ 内发生的所有事件,该顶点的每个边相关的事件包括关于该边的在时刻 $s_c$ 处有效的事件和在时间区间 $[s_c, t_c]$ 内发生的所有事件,基于该数据块内部索引,给定顶点ID,能够定位到在该空间-时间数据块中与该顶点相关的具体数据段。

15. 根据权利要求12的图数据管理方法, 计算某时刻进行图的时间分割的空间因子和时间因子, 基于空间因子和时间因子来确定进行时间分割的时刻, 所述图的时间分割包括在该时刻, 将未封印的空间-时间数据块变为封印的空间-时间数据块。

16. 根据权利要求12的图数据管理方法, 对于一个普通的未封印的空间-时间数据块, 设 $S_0$ 为该空间-时间数据块的开始时刻的快照的大小,  $L_u$ 为从所述开始时刻起到目前为止, 存储的空间-时间数据块日志的大小, 当 $L_u/S_0 \geq \lambda$ 且 $L_u \geq \gamma$ 时, 将该普通的未封印的空间-时间数据块转变为封印的空间-时间数据块, 其中 $\lambda$ 为类底数参数,  $\gamma$ 为最小的分割阈值。

17. 根据权利要求16的图数据管理方法, 在将该普通的未封印的空间-时间数据块转变为封印的空间-时间数据块时, 判断结束时刻 $t(n+1)$  e时的快照的大小是否超过预定阈值, 当超过预定阈值时, 将空间-时间数据块 $C(n+1)$ 的顶点集合分割为两个不相交的第一顶点集合和第二顶点集合, 以便关于第一顶点集合和第二顶点集合分别形成对应的快照和日志, 进而形成各自的空间-时间数据块。

18. 根据权利要求12-17任一项的图数据管理方法, 未封印的空间-时间数据块由快照和多个日志带组成, 日志带通过如下操作形成:

接收有关未封印的空间-时间数据块中的顶点的新发生的事件, 当累积的事件数据达到预定大小时, 将其形成为第一级别的日志带, 并继续接收新发生的事件以及形成第一级别的日志带的过程, 同时当第一级别的日志带的数目达到预定数目时, 将第一级别的日志带合并为第二级别的日志带, 并且当第二级别的日志带的数目达到预定数目时, 将第二级别的日志带合并为第三级别的日志带, 并重复此过程, 其中每个日志带都具有自己的索引, 当进行日志带合并时, 同时进行索引的合并。

19. 根据权利要求12-17任一项的图数据管理方法, 在形成封印空间-时间数据块时或封印空间-时间数据块已形成后, 重新确定各个顶点的数据段进行物理存储的顺序, 并按照所确定的顺序存储各个顶点的数据段。

## 时序图的图数据管理方法及其装置

### 技术领域

[0001] 本发明总体地涉及图的数据管理技术,更具体地涉及时序图的图数据的组织、存储、更新和查询技术。

### 背景技术

[0002] 图是一种由顶点的集合和顶点之间的关联关系即边的集合共同形成的数据结构。图也可以视为一种网络。现实生活中的许多问题,例如,社交网络中用户之间的关系、万维网中网页之间的关系、用户-项目矩阵(user-item matrix)、道路网络、语义网络等,都可以转化为图计算问题。

[0003] 这些图会随着时间不断演化,它们的变化规律是非常值得研究的问题。本文中将根据时间变化的图,称为时序图(temporary graph)。时序图分析研究时序图在一系列时间点上的快照,有的需要在这些快照上做全局迭代式计算,有的需要访问一些特定的顶点和边,以及它们的附属数据。典型的研究工作包括分析Web页面重要程度的变化过程、探索用户活动对他们在社交网络中的关系的影响,以及观察社交网络的图直径的变化等。时序图分析能够发现图在演化过程中体现出来的性质,增强了静态图分析的功能,正在成为图分析的重要手段。

[0004] 为了支持图分析工作的开展,从系统层面需要有图数据管理系统和图计算引擎的支持。对于时序图分析来说,引入时间维度后对系统支持提出了新的挑战,包括如何权衡空间和时间复杂度、如何利用数据局部性,因此需要设计新的系统来解决这些问题。对于计算部分,本文将在第4章详细讨论,本章将讨论时序图数据管理的问题,重点研究如何设计使得系统能够高效地支持时序图数据的存储和查询。

[0005] 在已有的相关工作中,DeltaGraph设计了一种类树结构用来存储时序图数据,其在获取快照的代价方面和有关顶点的随机访问开销上存在需要改进之处。

### 发明内容

[0006] 鉴于上述情况,做出了本发明。

[0007] 根据本发明的一个方面,提供了一种时序图的图数据管理方法,图由顶点和边组成,该图数据管理方法可以包括:获得时序图的事件数据;以及以二维空间-时间数据块 $C=(V_c, T_c)$ 形式组织时序图的数据并存储在存储设备上,一个维度是时间维度,另一个维度是顶点维度,数据块 $C=(V_c, T_c)$ 保存一个时间区间 $[s_c, t_c]$ 中与顶点集合 $V_c$ 相关的数据,所述数据块 $C=(V_c, T_c)$ 逻辑上包括与顶点集合 $V_c$ 相关的、在时刻 $s_c$ 处图的快照以及在时间区间 $[s_c, t_c]$ 内发生事件的日志,其中 $V_c$ 是顶点集合, $T_c$ 指示时间区间, $T_c=[s_c, t_c]$ , $s_c$ 表示该时间区间的起始时刻, $t_c$ 表示该时间区间的结束时刻。

[0008] 根据该图数据管理方法,与顶点集合 $V_c$ 相关的、在时刻 $s_c$ 处图的快照逻辑上可以以下列两种形式之一表示:在时刻 $s_c$ 处点和边的连接情况和属性设置情况;在时刻 $s_c$ 处有效的事件的数据的集合。

[0009] 图数据管理方法还可以包括:维持全局数据块索引,基于该全局数据块索引,给定顶点ID和时刻,能够定位到与该顶点ID和该时刻相关的数据在哪个数据块。

[0010] 空间-时间数据块还可以包括数据块内部索引,在基于给定顶点ID和时刻、基于全局数据块索引定位到具体空间-时间数据块后,基于数据块内部索引,基于该给定顶点ID,能够定位到在该具体空间-时间数据块中与该给定顶点ID相关的具体数据段。

[0011] 空间-时间数据块 $C = (V_c, T_c)$ 物理上可以包括多个顶点的数据段,每个顶点的数据段包含该顶点的事件以及所有以该顶点作为端点的边相关的事件,所述顶点的事件包括关于该顶点的、在时间 $s_c$ 处有效的事件和在时间区间 $[s_c, t_c]$ 内发生的所有事件,该顶点的每个边相关的事件包括关于该边的、在时间 $s_c$ 处有效的事件和在时间区间 $[s_c, t_c]$ 内发生的所有事件,每个事件信息包括事件内容和事件发生的时刻。

[0012] 多个顶点的数据段在物理上可以是一个顶点的数据段一个顶点的数据段地顺序存储的,每个顶点的数据段中的所有数据集中存储在一起。

[0013] 图数据管理方法还可以包括,在顶点维度上,针对已经存在的每个顶点集合,时间上从该顶点集合的最后一个空间-时间数据块 $C_n$ 的结束时刻 $t_{ne}$ 开始,存储结束时刻 $t_c$ 的快照作为与该顶点集合相关联的最新快照,以及随时间进行,接收新发生的事件,并以日志形式存储。

[0014] 9、根据权利要求8的图数据管理方法,还包括:

[0015] 关于该顶点集合,以最后一个空间-时间数据块 $C_n$ 的结束时刻 $t_{ne}$ 时刻作为下一空间-时间数据块 $C_{n+1}$ 的开始时刻 $t_{(n+1)s}$ ,确定下一空间-时间数据块 $C_{(n+1)}$ 的结束时刻 $t_{(n+1)e}$ ,所述下一空间-时间数据块 $C_{(n+1)}$ 的时间区间 $[t_{(n+1)s}, t_{(n+1)e}]$ 的长度是目前数据块的总数 $k$ 的指数函数;以及基于开始时刻 $t_{(n+1)s}$ 的快照和所述从所述开始时刻 $t_{(n+1)s}$ 起到目前为止、关于最后一个空间-时间数据块 $C_n$ 的顶点集合存储的日志,形成下一空间-时间数据块 $C_{(n+1)}$ 。

[0016] 在图数据管理方法中,关于该顶点集合,可以以最后一个空间-时间数据块 $C_n$ 的结束时刻 $t_{ne}$ 时刻作为下一空间-时间数据块 $C_{n+1}$ 的开始时刻 $t_{(n+1)s}$ ,如下确定下一空间-时间数据块 $C_{(n+1)}$ 的结束时刻 $t_{(n+1)e}$ :设 $S_u$ 为下一空间-时间数据块 $C_{(n+1)}$ 的开始时刻 $t_{(n+1)s}$ 的快照的大小, $L_u$ 为从所述开始时刻 $t_{(n+1)s}$ 起到目前为止、关于最后一个空间-时间数据块 $C_n$ 的顶点集合存储的日志的大小,当 $L_u/S_u \geq \lambda$ 且 $L_u \geq \gamma$ 时,基于开始时刻 $t_{(n+1)s}$ 的快照和所述从所述开始时刻 $t_{(n+1)s}$ 起到目前为止、关于最后一个空间-时间数据块 $C_n$ 的顶点集合存储的日志,形成下一空间-时间数据块 $C_{(n+1)}$ ,其中 $\lambda$ 为类底数参数, $\gamma$ 为最小的分割阈值。

[0017] 图数据管理方法还可以包括:形成下一空间-时间数据块 $C_{(n+1)}$ 时,判断下一空间-时间数据块 $C_{(n+1)}$ 的结束时刻 $t_{(n+1)e}$ 的快照的大小是否超过预定阈值,当超过预定阈值时,将空间-时间数据块 $C_{(n+1)}$ 的顶点集合分割为两个不相交的第一顶点集合和第二顶点集合,后续将关于第一顶点集合和第二顶点集合分别形成对应的快照和日志,进而形成各自的空间-时间数据块。

[0018] 图数据管理方法还可以包括;在形成空间-时间数据块时或空间-时间数据块已形成后,重新确定各个顶点的数据段进行物理存储的顺序,并按照所确定的顺序存储各个顶点的数据段。

[0019] 图数据管理方法还可以包括:对于新出现的不属于任何已经存在的顶点集合的顶点,维持一最新的顶点集合,最新的顶点集合中的顶点不属于任何空间-时间数据块相关联的顶点集合,将随时间进行新出现的顶点添加入该最新的顶点集合,并以日志形式存储关于该最新顶点集合中各顶点的事件数据,当该日志数据大小超过预定阈值时,基于此时的最新顶点集合和相关联的时间区间,形成空间-时间数据块;以及将最新顶点集合清零,并接收后续出现的新顶点和相关联的事件数据。

[0020] 图数据管理方法还可以包括:接收查询,该查询涉及给定顶点或给定顶点集合和涉及给定时刻或给定时间区间;基于该给定顶点或给定顶点集合和给定时刻或给定时间区间,查询全局数据块索引,定位到相关联的空间-时间数据块;对于定位到的相关联的空间-时间数据块的每个,查询与该空间-时间数据块相关联的数据块内部索引,定位到与给定顶点或给定顶点集合相关联的具体数据段;以及扫描该具体数据段,返回该具体数据段的查询结果;以及合并各个查询结果,并返回合并得到的查询结果。

[0021] 根据本发明的另一方面,提供了一种图数据管理装置,可以包括:事件摄入部件,配置为摄入新发生的事件,并发送至时序图数据管理引擎;查询引擎,配置为接收来自外部的查询,并将该查询发送至时序图数据管理引擎,接收来自时序图数据管理引擎的查询结果,并输出该查询结果;以及时序图数据管理引擎,以二维空间-时间数据块 $C = (V_c, T_c)$ 形式组织时序图的数据并存储在存储设备上,一个维度是时间维度,另一个维度是顶点维度,数据块 $C = (V_c, T_c)$ 保存一个时间区间 $[s_c, t_c]$ 中与顶点集合 $V_c$ 相关的数据,所述数据块 $C = (V_c, T_c)$ 逻辑上包括在时刻 $s_c$ 处图的快照以及在时间区间 $[s_c, t_c]$ 内发生事件的日志,其中 $V_c$ 是顶点集合, $T_c$ 指示时间区间, $T_c = [s_c, t_c]$ , $s_c$ 表示该时间区间的起始时刻, $t_c$ 表示该时间区间的结束时刻,在时刻 $s_c$ 处图的快照包括在时刻 $s_c$ 处有效的事件的数据的集合。

[0022] 根据本发明的另一方面,提供了一种时序图的图数据管理方法,图由顶点和边组成,该图数据管理方法可以包括:以二维空间-时间数据块 $C = (V_c, T_c)$ 形式组织时序图的数据并存储在存储设备上,一个维度是时间维度,另一个维度是顶点维度,数据块 $C = (V_c, T_c)$ 保存一个时间区间 $[s_c, t_c]$ 中与顶点集合 $V_c$ 相关的数据,所述数据块 $C = (V_c, T_c)$ 逻辑上包括在时刻 $s_c$ 处图的快照以及在时间区间 $[s_c, t_c]$ 内发生事件的日志,其中 $V_c$ 是顶点集合, $T_c$ 指示时间区间, $T_c = [s_c, t_c]$ , $s_c$ 表示该时间区间的起始时刻, $t_c$ 表示该时间区间的结束时刻,在时刻 $s_c$ 处图的快照包括在时刻 $s_c$ 处有效的事件的数据的集合;将空间-时间数据块分类为封印的空间-时间数据块和未封印的空间-时间数据块,对于基于同样的顶点集合的封印的空间-时间数据块和未封印的空间-时间数据块,未封印的空间-时间数据块时间上晚于封印的空间-时间数据块,且保持开放状态以等待接收关于该顶点集合中的顶点的任何新发生的事件,而封印的空间-时间数据块已封闭不再接收新的事件;对于任一未封印的空间-时间数据块,当满足预定的分割条件时,以满足预定的分割条件的时刻作为分界点,将该未封印的空间-时间数据块转变为新的封印的空间-时间数据块,该新的封印的空间-时间数据块以该分界点作为结束时刻,创建新的未封印的空间-时间数据块,该新的未封印的空间-时间数据块以该分界点为开始时刻,该新的封印的空间-时间数据块与该新的未封印的空间-时间数据块具有相同的顶点集合;以及将未封印的空间-时间数据块分类为普通的未封印的空间-时间数据块和特殊的未封印的空间-时间数据块,普通的未封印的空间-时间数据块的顶点集合出现于封印的空间-时间数据块的顶点集合,特殊的未封印的空间-时



间数据块的顶点集合中的顶点不曾出现于任何封印的空间-时间数据块,在任何当前时刻维持一个特殊的未封印的空间-时间数据块,当出现的事件涉及新顶点时,将该事件添加入该特殊的未封印的空间-时间数据块,当出现的事件涉及存在于普通的未封印的空间-时间数据块的顶点集合中的顶点时,将该事件添加入该普通的未封印的空间-时间数据块。

[0023] 图数据管理方法还可以包括:维持全局数据块索引和数据块内部索引,基于全局数据块索引,给定顶点ID和时刻,能够定位到具体空间-时间数据块,基于数据块内部索引,能够定位到在该具体数据块中与该顶点相关的具体数据段。

[0024] 空间-时间数据块 $C = (V_c, T_c)$ 物理上可以包括顺序存储多个顶点的数据段和包括数据块内部索引,每个顶点的数据段包含该顶点的事件以及所有该顶点的边相关的事件,所述顶点的事件包括关于该顶点的、在时间 $s_c$ 处有效的事件和在时间区间 $[s_c, t_c]$ 内发生的所有事件,该顶点的每个边相关的事件包括关于该边的、在时间 $s_c$ 处有效的事件和在时间区间 $[s_c, t_c]$ 内发生的所有事件,基于该数据块内部索引,给定顶点ID,能够定位到在该数据块中与该顶点相关的具体数据段。

[0025] 图数据管理方法可以包括计算某时刻进行图的时间分割的空间因子和时间因子,基于空间因子和时间因子来确定进行时间分割的时刻,所述图的时间分割包括在该时刻,将未封印的空间-时间数据块变为封印的空间-时间数据块。

[0026] 在一个示例中,对于一个普通的未封印的空间-时间数据块,设 $S_u$ 为该空间-时间数据块的开始时刻的快照的大小, $L_u$ 为从所述开始时刻起到目前为止,存储的空间-时间数据块日志的大小,当 $L_u/S_u \geq \lambda$ 且, $L_u \geq \gamma$ 时,将该普通的未封印的空间-时间数据块转变为封印的空间-时间数据块,其中 $\lambda$ 为类底数参数, $\gamma$ 为最小的分割阈值。

[0027] 在图数据管理方法中,可以在将该普通的未封印的空间-时间数据块转变为封印的空间-时间数据块时,判断结束时刻 $t(n+1)$ 时的快照的大小是否超过预定阈值,当超过预定阈值时,将空间-时间数据块 $C(n+1)$ 的顶点集合分割为两个不相交的第一顶点集合和第二顶点集合,以便关于第一顶点集合和第二顶点集合分别形成对应的快照和日志,进而形成各自的空间-时间数据块。

[0028] 未封印的空间-时间数据块可以由快照和多个日志带组成,日志带可以通过如下操作形成:接收有关未封印的空间-时间数据块中的顶点的新发生的事件,当累积的事件数据达到预定大小时,将其形成为第一级别的日志带,并继续接收新发生的事件以及形成第一级别的日志带的过程,同时当第一级别的日志带的数目达到预定数目时,将第一级别的日志带合并为第二级别的日志带,并且当第二级别的日志带的数目达到预定数目时,将第二级别的日志带合并为第三级别的日志带,并重复此过程,其中每个日志带都具有自己的索引,当进行日志带合并时,同时进行索引的合并。

[0029] 可以在形成封印空间-时间数据块时或封印空间-时间数据块已形成后,重新确定各个顶点的数据段进行物理存储的顺序,并按照所确定的顺序存储各个顶点的数据段。

[0030] 根据本发明实施例的时序图的图数据管理方法和图数据管理装置,以二维空间-时间数据块形式组织时序图,适合于高效地存储和查询时序图数据。

[0031] 根据本发明另一实施例的时序图的图数据管理方法,以封印空间-时间数据块、普通未封印空间-时间数据块、特殊未封印空间-时间数据块来组织时序图,能够很好地组织各种顶点和边事件,顺利进行未封印空间-时间数据块向封印空间-时间数据块的转换,为

查询操作获得较为平衡的空间和时间开销。

### 附图说明

[0032] 从下面结合附图对本发明实施例的详细描述中,本发明的这些和/或其它方面和优点将变得更加清楚并更容易理解,其中:

[0033] 图1示出了示例性的时序图演化过程1000。

[0034] 图2示出了根据本发明实施例的时序图的图数据管理方法2000的流程图。

[0035] 图3示出了根据本发明实施例的以二维空间-时间数据块形式组织时序图的示例性示意图3000。

[0036] 图4示出了根据本发明实施例的以二维空间-时间数据块形式组织时序图的示例性示意图4000。

[0037] 图5示出了根据本发明实施例的全局数据块索引的数据结构的形式以及基于全局数据块索引定位具体空间-时间数据块的示意图5000。

[0038] 图6示出了根据本发明实施例的空间-时间数据块的数据结构6000的示意图。

[0039] 图7示出了由起始时刻快照结合多个日志带组成的空间-时间数据块的数据布局的示意图7000。

[0040] 图8示出了根据本发明实施例的未封印的数据块的形成过程示意图8000。

[0041] 图9示出了不同的顶点的数据段在磁盘上的排列顺序对影响基于遍历的查询的示意图。

[0042] 图10示出了根据本发明实施例的时序图的特定时刻t的全局查询方法10000的流程图。

[0043] 图11示出了根据本发明实施例的局部查询方法11000的流程图。

[0044] 图12示出了根据本发明实施例提供的图数据管理装置12000的结构示意图。

### 具体实施方式

[0045] 为了使本领域技术人员更好地理解本发明,下面结合附图和具体实施方式对本发明作进一步详细说明。

[0046] 在进行详细说明之前,对本文中一些术语的含义进行说明。

[0047] 时序图的快照:即,某个时刻的图,可以视为时间轴上时序图在某指定时刻的横切面。

[0048] 事件:指对图所做的改变,例如顶点的添加、删除和顶点属性的设置和删除,以及边的添加、删除和边属性的设置和删除。

[0049] 某时刻的有效事件:对该时刻的图的可视化状态起直接的积极性(或建设性)作用的事件,有效事件直接决定了该时刻点和/或边的存在以及存在状态。

[0050] 全局查询:全局查询经常用在图计算中,用于获取要进行计算的图快照。全局查询访问时序图在给定时刻t的快照中存在的所有顶点和边。尽管全局操作可以由访问每个顶点的局部操作拼成,出于效率考虑,一般需要系统提供单独的全局查询接口。

[0051] 局部查询:局部查询访问时序图在给定时刻t的快照中某个顶点v和它所有的边,同时也可以访问这些边所指向的顶点(邻居顶点)。局部查询只访问在时刻t存在的顶点或

边。更加复杂的时序图查询可以用局部查询来实现。例如,一个顶点的二阶邻居顶点(比如社交网络中一个用户朋友的朋友)可以由一个局部查询接一组局部查询来实现,后一组局部查询的顶点是第一个查询结果中的所有邻居顶点。

[0052] 图1示出了示例性的时序图演化过程1000。其中示出了分别在时刻 $t_0$ 、 $t_1$ 、 $t_2$ 、 $t_3$ 、 $t_4$ 的图的快照 $G_0$ - $G_4$ ,以及在下面注明了发生的事件。在时刻 $t_0$ ,为快照 $G_0$ ;在时刻 $t_1$ ,快照为图 $G_1$ ,事件为 $(AV, v_0, 1)$ 和 $(AV, v_1, 1)$ ,即在时刻 $t_1$ 添加顶点 $v_0$ 和 $v_1$ ;在时刻 $t_2$ ,快照为 $G_2$ ,事件为 $(AV, v_2, 2)$ 和 $(AE, e_0, v_1, v_0, 2)$ ,即在时刻 $t_2$ 增加了顶点 $v_2$ 以及增加了边 $e_0$ , $e_0$ 的起始顶点为 $v_1$ ,结束顶点为 $v_0$ ;在时刻 $t_3$ ,快照为 $G_3$ ,事件为 $(AE, e_1, v_0, v_1, 3)$ 和 $(AE, e_2, v_2, v_0, 3)$ ,即在时刻 $t_3$ 增加了边 $e_1$ , $e_1$ 的起始顶点为 $v_0$ ,结束顶点为 $v_1$ ,以及增加了边 $e_2$ , $e_2$ 的起始顶点为 $v_2$ ,结束顶点为 $v_0$ ;在时刻 $t_4$ ,快照为 $G_4$ ,事件为 $(RE, e_1, 4)$ ,即在时刻 $t_4$ 去除边 $e_1$ 。

[0053] 图1所示的时序图仅为示例,旨在说明随时间图的演化、图的快照以及事件的概念。当然随着应用的不同,图的形式、复杂程度、变化情况可以不同。

[0054] 下面结合图2描述根据本发明实施例的时序图的图数据管理方法。图2示出了根据本发明实施例的时序图的图数据管理方法2000的流程图。

[0055] 如图2所示,在步骤S2100中,获得时序图的图数据。

[0056] 这里获得时序图的图数据,是广义上的。可以是实时地获得事件并缓存,待达到一定数量后传给后面的步骤S2200。也可以是从外部获得以其他形式组织的时序图数据,所述其他形式组织的例子包括:例如,完全的快照式组织,即将每一时刻的快照存储下来;另一种是,例如完全的日志式组织,即只存储各个事件的日志,且每个事件只存储一次。

[0057] 在步骤S2200中,以二维空间-时间数据块 $C = (V_c, T_c)$ 形式组织时序图的数据并存储在存储设备上,一个维度是时间维度,另一个维度是顶点维度,数据块 $C = (V_c, T_c)$ 保存一个时间区间 $[s_c, t_c]$ 中与顶点集合 $V_c$ 相关的数据,所述数据块 $C = (V_c, T_c)$ 逻辑上包括与顶点集合 $V_c$ 相关的、在时刻 $s_c$ 处图的快照以及在时间区间 $[s_c, t_c]$ 内发生事件的日志,其中 $V_c$ 是顶点集合, $T_c$ 指示时间区间, $T_c = [s_c, t_c]$ , $s_c$ 表示该时间区间的起始时刻, $t_c$ 表示该时间区间的结束时刻。

[0058] 图3示出了根据本发明实施例的以二维空间-时间数据块形式组织时序图的示例性示意图3000。竖轴 $V$ 示意顶点维度,其是离散的;横轴 $t$ 表示时间维度,其本质上是连续的,但也可以人为设置为离散形式。图3中示出了截止至当前时刻 $t_c$ 所形成的空间-时间数据块 $C_0$ 、 $C_1$ 、 $C_2$ 、 $C_3$ 、 $C_4$ 。其中 $C_0$ 顶点维度为顶点集合 $V_1$ ,时间维度跨越时间区间 $[0, s_1]$ ;  $C_1$ 顶点维度为顶点集合 $V_1$ ,时间维度跨越时间区间 $[s_1, t_1]$ ;  $C_2$ 顶点维度为顶点集合 $V_2$ ,时间维度跨越时间区间 $[s_1, t_2]$ ;  $C_3$ 顶点维度为顶点集合 $V_{11}$ ,时间维度跨越时间区间 $[t_1, t_3]$ ;其中 $C_4$ 顶点维度为顶点集合 $V_{12}$ ,时间维度跨越时间区间 $[t_1, t_3]$ 。

[0059] 图4示出了根据本发明另一实施例的以二维空间-时间数据块形式组织时序图的示例性示意图4000。图4与图3的不同在于,除了空间-时间数据块 $C_0$ 、 $C_1$ 、 $C_2$ 、 $C_3$ 、 $C_4$ 外,还包括空间-时间数据块 $U_0$ 到 $U_4$ 。空间-时间数据块 $U_0$ 到 $U_4$ 相比于空间-时间数据块 $C_0$ 到 $C_4$ 的区别在于:空间-时间数据块 $U_0$ 到 $U_4$ 处于开放状态,随时等待接收新的事件,其时间区间的起始点已确定,但终点尚未确定;而空间-时间数据块 $C_0$ 到 $C_4$ 则处于封印状态,不再接收新的事件,时间区间的起始点和终点都已确定。空间-时间数据块 $U_0$ 到 $U_4$ 和空间-时间数据块 $C_0$ 到 $C_4$ 的共同之处在于逻辑上都存储有开始时刻的快照和从开始时刻起的时间区间的事件日志。在后

文中,将空间-时间数据块 $C_0$ 到 $C_4$ 等称为封印数据块,将空间-时间数据块 $U_0$ 到 $U_4$ 等称为未封印数据块。在未封印数据块 $U_0$ 到 $U_4$ 中,未封印数据块 $U_4$ 是特殊的未封印数据块,其快照是空的,因为其内存的新数据都是关于新顶点的,该新顶点是在 $U_4$ 的初始时刻之后由新出现的添加顶点事件产生的,即特殊的未封印数据块 $U_4$ 中涉及的顶点不曾出现于任何封印的空间-时间数据块中。

[0060] 与顶点集合 $V_c$ 相关的、在时刻 $S_c$ 处图的快照可以以下述第一种形式表达:在时刻 $S_c$ 处点和边的连接情况和属性设置情况,例如对于各个顶点,存储该顶点的属性,关于该顶点的边的连接,以及边的属性。

[0061] 在一个示例中,与顶点集合 $V_c$ 相关的、在时刻 $S_c$ 处图的快照可以以下述第二种形式表达:在时刻 $S_c$ 处有效事件的数据的集合,有效事件的类型包括点的添加、边的添加、点的属性的设置、边的属性的设置,之所以称之为有效事件,是因为该事件直接决定了 $S_c$ 时刻点和/或边的存在以及存在状态。例如,在 $S_c$ 时刻之前,按时间先后顺序发生了事件1(设置特定边的某属性为 $a_1$ )、事件2(设置该特定边的该属性为 $a_2$ )、事件3(设置该特定边的该属性为 $a_3$ ),则事件3(设置该特定边的该属性为 $a_3$ )是在 $S_c$ 时刻的有效事件,而事件1、事件2则在 $S_c$ 时刻失效了,可称为失效的事件。再例如,在 $S_c$ 时刻之前,按时间先后顺序发生了事件1(设置特定边的某属性为 $a_1$ )、事件2(设置该特定边的该属性为 $a_2$ )、事件3(设置该特定边的该属性为 $a_3$ )、事件4(删除该特定边),则事件1-4都不是 $S_c$ 时刻的有效事件,因为在 $S_c$ 时刻图的可视化结构中根本看不见这条边的存在。每个事件的数据包括事件的内容和事件发生的时刻。

[0062] 快照的第二种形式表达相对于第一种形式表达多出了有效事件发生的时间信息,在某些应用中,需要用到有效事件的时间信息,例如在社交网络facebook应用中统计当前用户中在2009年前注册的用户个数,则此时用户注册事件的时间信息就是重要的。

[0063] 下面的例子中,将以时刻 $S_c$ 处的图的快照以有效事件的集合形式表达为例加以描述。

[0064] 在一个示例中,维持全局数据块索引(也可称之为全局索引,两者可互换使用),基于该全局数据块索引,给定顶点ID和时刻,能够定位到与该顶点ID和该时刻相关的数据在哪个数据块。

[0065] 在一个示例中,数据定位索引包括全局数据块索引和数据块内部索引,基于全局数据块索引,给定顶点ID和时刻,能够定位到具体数据块,基于数据块内部索引,能够定位到在该具体数据块中与该顶点相关的具体数据段。

[0066] 图5示出了根据本发明实施例的全局数据块索引的数据结构的形式以及基于全局数据块索引定位具体空间-时间数据块的示意图5000。

[0067] 如图5所示,全局数据块索引5100包括多个全局索引项5110,每个全局索引项5110的关键字为顶点id和时间。全局索引项5110可以包括多个域,包括顶点id、时间、数据块id、偏移,这样基于顶点id和时间,能够确定相应的数据块id,并依据偏移,而定位到该数据块id指示的数据块,偏移指示该数据块id指示的数据块相对于数据块开头的偏移地址。

[0068] 在一个示例中,空间-时间数据块还包括数据块内部索引,在基于给定顶点ID和时刻、基于全局数据块索引定位到具体空间-时间数据块后,基于数据块内部索引,基于该给定顶点ID,能够定位到在该具体空间-时间数据块中与该给定顶点ID相关的具体数据段。

[0069] 图6示出了根据本发明实施例的空间-时间数据块的数据结构6000的示意图。

[0070] 在图6所示的示例中,空间-时间数据块的数据结构6000包括数据块内部索引6100和多个顶点的数据段6200。数据块内部索引6100包括多个索引项,每个索引项包括顶点ID和偏移,偏移信息可以包括指示该顶点ID相关的数据段6200的起始存储位置的信息,还可以指示该顶点ID相关的数据段6200的大小或结束位置的信息。多个顶点的数据段6200包括顶点 $v_0$ 的数据段6200、顶点 $v_1$ 的数据段6200、顶点 $v_2$ 的数据段6200等等。顶点 $v_0$ 的数据段6200包括顶点的事件以及所有该顶点的边相关的事件。根据应用的不同,以该顶点为端点的边,可以是该顶点的出边、该顶点的入边或者入边和出边两者。在下面的示例中,将以关注的边为出边为例加以说明,不过这仅为示例,而非作为对本发明的限制,随应用的不同关注的边的类型可以不同。如图6所示,顶点 $v_0$ 的数据段包含顶点 $v_0$ 的事件6210以及所有 $v_0$ 的出边相关的事件6220。

[0071] 每个顶点的事件6210和每个边的事件6220可以共享相同的数据结构,可以有若干个域,每个域例如可以用64位整数表示。

[0072] • 顶点id 6211/边id 6221:每个顶点的id6211是全局唯一的。最高的3位被保留起来,作为标志位,分别表示这个事件是顶点的事件还是边的事件、是添加事件(或者设置事件)还是删除事件,以及是关于实体的事件(也就是图的拓扑结构的变化)还是关于属性的事件。在边的事件的情况下,如果应用只关注出边,可以设定边的id6221以其起点的id为前缀。

[0073] • 时间6212或6222:事件发生时的时刻。在一个示例中,可以用UNIX时间来表示,以例如毫秒为单位。

[0074] • 终点id6223(可选):当事件为边的添加时,该域表示边的终点id。

[0075] • 数据6213或6224(可选):例如,由4个子域组成,分别是32位的关键字长度、32位的值长度、关键字内容和值的内容,用于描述属性的变化。在一个示例中,关键字和值的内容可以是变长的字节数组,填充并对齐到8个字节。

[0076] 图6所示的每个顶点的事件6210中位于数据6213后的省略号表示后边是关于顶点 $v_0$ 的其它事件,即是重复的域6211-6213;每个边的事件6220中位于数据6224后的省略号表示后边是关于相同边的其它事件,即是重复的域6221-6224。另外顶点 $v_0$ 的数据段示出了多个边的事件,是因为一个顶点可以涉及多条边,分别表示关于该顶点的边同边的事件。

[0077] 需要说明的是,图6所示的空间-时间数据块的结构仅为示例,并非作为本发明的限制。可以根据需要做出改变,例如,对于顶点的事件,可以不是为特定顶点的每个事件都分配顶点ID比特,而是该特定顶点的所有事件共享相同的顶点ID域,此时,该特定顶点的每个事件数据可以由标志位、时间、数据(可选)形成。

[0078] 这里需要说明的是,与顶点集合 $V_c$ 相关的、在时刻 $s_c$ 处图的快照,并不必然意味着将描述与顶点集合 $V_c$ 相关的、在时刻 $s_c$ 处图的快照的数据物理上统一存储在一起,而是可以是该数据分散到各个顶点的数据段中,如图6所示的那样。也就是说,在图6所示的数据结构中,顶点 $v_0$ 的数据段中,关于顶点 $v_0$ 的在时刻 $s_c$ 处图的快照(时刻 $s_c$ 处有效事件的集合)的数据是分散在 $v_0$ 的事件、边的事件中的,例如可以将 $v_0$ 的时刻 $s_c$ 处有效事件放置在 $v_0$ 的所有事件的最前端,关于 $v_0$ 的边的时刻 $s_c$ 处有效事件放置在关于 $v_0$ 的相应边的所有事件的最前端,等等。

[0079] 在一个示例中,图6所示的各个顶点的数据段6200在物理上是顺序存储的。每个顶点的数据段中的所有数据集中存储在一起,一个顶点的数据段一个顶点的数据段地顺序存储。

[0080] 本发明实施例的空间-时间数据块组织形式非常适合于封印的空间-时间数据块,以及有利于进行全局查询和局部查询。当一个查询涉及到顶点集合 $V_c$ 中的顶点,且查询时刻在 $T_c$ 区间内时,查询结果需要用到数据块C中的数据。进行时刻 $t$ 的全局查询可以通过扫描数据块C来获得查询结果的过程,该算法顺序地扫描整个数据块,跳过发生时刻大于 $t$ 的所有事件。对于任意的一个顶点、一条边,或者一个属性,算法只输出在时刻 $t$ 之前或正好发生在时刻 $t$ 的最后一个相关的事件。其他更早的事件由于其状态被后来的事件更新了,因此不需要输出。另外,如果发现最新的一个事件是删除事件,则该对象在时刻 $t$ 已被删除,同样也不需要输出。

[0081] 在局部查询的情况下,如果对顶点 $v$ 在时刻 $t$ 的局部查询落在数据块C中,也就是同时满足 $v \in VC$ 和 $t \in TC$ ,则查询结果都在数据块C的一个数据段内。因此,局部查询只需要对相关的数据段做一次随机I/O访问(另外还有对全局数据块索引的访问)。封印数据块的布局使得局部查询的I/O开销最小。

[0082] 在另一示例中,一个空间-时间数据块的数据布局可以作为一个物理上集中存储的开始时刻的快照、结合多个日志带,其中该多个日志带中至少每个日志带的数据是物理上集中存储的,以及快照和每个日志带都具有类似于图6所示的数据结构,即快照和每个日志带都具有索引。图7示出了由起始时刻快照结合多个日志带组成的空间-时间数据块的数据布局的示意图7000。该空间-时间数据块的数据布局特别适合于未封印数据块。

[0083] 在一个示例中,未封印的空间-时间数据块中的日志带通过如下操作形成:接收有关未封印的空间-时间数据块中的顶点的新发生的事件,当累积的事件数据达到预定大小时,将其形成为第一级别的日志带,并继续接收新发生的事件以及形成第一级别的日志带的过程,同时当第一级别的日志带的数目达到预定数目时,将第一级别的日志带合并为第二级别的日志带,并且当第二级别的日志带的数目达到预定数目时,将第二级别的日志带合并为第三级别的日志带,并重复此过程。

[0084] 图8示出了根据本发明实施例的未封印的数据块的形成过程示意图8000。在一个示例中,随着时间的进行,将新来的事件插入到内存中的事件表里。当内存中的事件表大小增长到一定的阈值时,将该事件表写入到磁盘,变成一个日志带。日志带的格式可以与图6所示的封印数据块的格式相同。在一个示例中,将日志带根据大小分为不同的级别,最低的级别为1级。如果磁盘上同时存在 $k$ 个大小相同的1级日志带时,将它们合并为一个2级日志带。当这个2级日志带写完后,原来的 $k$ 个以及日志带就可以被丢弃了。以此类推,当磁盘上存在 $k$ 个 $j$ 级日志带时,它们会被合并为一个 $j+1$ 级日志带。整数 $k$ 的大小可以根据需要设置。

[0085] 之所以在未封印数据块形成数据块过程中进行这样的日志带逐级合并的处理,是因为日志带合并操作能够为全局查询和局部查询方面带来性能提升。未封印数据块 $U$ 上时刻 $t$ 的查询需要访问快照、与时间区间 $[s_u, t]$ 有交集的日志带,如果内存中的事件表也包含 $[s_u, t]$ 内发生的事件,则也要访问。全局查询可以用与上述对于封印数据块对全局查询计算快照的算法类似的算法来计算时刻 $t$ 的快照。局部查询则需要先查出所有相关部分(快照、日志带、内存中的事件表)中所需数据段的偏移量,然后再把这些数据段合并起来产生

最后的结果。从上面的讨论可以看出日志带合并的好处。首先,合并操作使得全局查询所需访问的日志带的个数有了一定的限制。其次,局部查询很有可能要为每一个数据段进行一次随机I/O访问,减少日志带的个数也就减少了随机I/O的次数。

[0086] 在一个示例中,随着新的事件不断加入,未封印数据块的大小越来越大。为了将数据组织成大小可控的单位,获得好的空间和时间效率,可以对未封印数据块进行分割,分割可以包括时间分割和图分割(或空间分割)。时间分割是指某个时间点 $t$ 将未封印数据块分为两部分,将前一部分转变为封印数据块,而后面一部分则变成新的未封印数据块,它们的顶点集合相同(如图4中的 $C_2$ 和 $U_3$ )。图分割操作则把数据块按照顶点集合一分为二。在一个示例中,简化起见,可以只在做时间分割的同时选择做图分割。一个时间分割和图分割的组合操作会把一个未封印数据块变为一个封印数据块和两个未封印数据块(如图4中的 $C_3$ 、 $U_0$ 和 $U_1$ )。

[0087] 封印操作是指将未封印数据块转变为封印数据块的操作,包括将未封印数据块 $U$ 的快照、所有的日志带、以及内存中的事件表进行合并。合并操作包括:对于每个顶点,扫描快照、日志带和内存表中的事件表,将涉及该顶点的事件和该顶点的边的事件集中到一起,按照例如图6所示的形式进行组织,以及确定各个顶点的地址(即,确定偏移),生成新的数据块内部索引,最后更新全局数据块索引,使得基于顶点 $id$ 和时间,能够定位到该新生成的封印数据块。在新的未封印数据块 $U'$ 中,未封信数据块 $U$ 中所有到时刻 $t$ (即分割时间点,新形成的封印数据块的结束时刻,以及新的未封印数据块 $U'$ 的开始时刻)时已经失效的事件都会被去掉,所谓已经失效的事件是指对时刻 $t$ 的可视化图数据(涵盖拓扑结构和属性设置)没有效用的事件,例如,时刻 $t$ 时某边的属性值为 $Value_{te}$ ,是在时刻 $t_e$ 由边属性设置事件得到的结果,则在时刻 $t_e$ 前其它对边的属性值设置都是失效的事件;类似地,如果在时刻 $t$ 之前,关于某顶点的最后操作事件是删除,则关于该顶点的任何事件都是失效的事件,以及关于该顶点的边的事件也变成了失效的事件。对于有大量更新或删除事件的数据块,这样做可以显著地减小未封印数据块的大小 $U'$ 。在未封印数据块 $U$ 上做时刻 $t$ 的全局查询所得到的快照,就是新的未封印数据块 $U'$ 开始的快照。从时刻 $t$ 起,关于相关顶点的、所有新的事件都将追加到 $U'$ 。

[0088] 为了避免时间分割操作和在时间分割操作进行中新增加的事件之间的相互影响,时间分割操作进行中的新增事件将会被加入到一个新的日志带中。当系统完成 $C$ 和 $U'$ 的创建后,必须原子地从 $U$ 切换到 $C$ 和 $U'$ 。在切换过程中,全局数据块索引将会被锁定,指向新建数据块的索引项会加入到全局数据块索引中。这里,所谓“原子地”是指这个过程从系统其他部分来看是一个整体,中间不能打断,也不能让系统其他部分看到过程的中间状态。实现的时候需要加锁。这样,全局数据块索引中包括了指向新的封印数据块 $C$ 的全局索引项和包括指向新的未封印数据块 $U'$ 的全局索引项,老的未封印数据块 $U$ 就可以被丢弃了。时间分割操作完成后,早于时刻 $t$ 的查询都会在封印数据块 $C$ 上进行。封印数据块 $C$ 不包含任何发生时间在时刻 $t$ 或之后的事件。而对最近的时刻的查询将会在新的未封印数据块 $U'$ 上进行,在时刻 $t$ 之前就失效的事件已经被移除。因此,通过进行时间分割,查询操作的性能得到了提升。

[0089] 进行时间分割操作的时刻会关系到系统的性能。下面讨论如何选择时间分割点的问题。

[0090] 根据本发明的一个实施例,通过分析全局查询的时间开销和整体的空间开销,来

指导选取时间分割的策略。

[0091] 在一个示例中,定义空间因子SF为至特定时刻为止全部数据块所占大小与所有事件大小之比,以及定义时间因子TF为访问某时刻全局快照所要扫描的数据块的大小与该时刻全局快照的实际大小之比。考察在图恒定均匀增长的理想情况下,将时间分割点定义为时间分割编号n(或可以理解为封印数据块的总数)的多项式函数或指数函数(设指数函数的底数为b),分别考察对应的时间因子和空间因子。底数b为大于1的实数,其取值基于时间和空间的权衡,b越小占用空间越多,但是查询性能越好。由分析可以得出,空间和时间开销是一种权衡关系,更小的时间因子TF<sub>n</sub>对应着更大的空间因子SF<sub>n</sub>。经分析发现,当向图添加事件的频率大于0时,指数时间分割可以获得较为平衡的空间和时间开销;不过当向图添加事件的频率等于0时,指数时间分割的效果会变差。

[0092] 根据本发明的一个实施例,提出了自适应指数时间分割策略。设S<sub>u</sub>为最近一个快照的大小,L<sub>u</sub>为未封印数据块U中到目前为止的日志的大小,则自适应指数时间分割策略会在 $L_u/S_u \geq \lambda$ 且 $L_u \geq \gamma$ 时进行时间分割操作。其中, $\lambda$ 为类底数参数(与指数时间分割策略中的底数b有一定的关系,当时序图为理想的恒定增长图时, $\lambda = (b-1)/\alpha$ ), $\alpha$ 指示添加事件的频率, $\gamma$ 表示最小的分割阈值,即 $\gamma$ 指示最小进行时间分割的日志大小,防止数据块太小时做很多分割操作,在一个示例中 $\gamma$ 取值例如为64MB。与指数时间分割策略类似, $\lambda$ 也用于调节空间和时间因子的权衡关系。当 $\alpha \geq 0$ 时,自适应指数时间分割策略都能获得常数的空间和时间因子。

[0093] 现有的时序图管理算法DeltaGraph中的快照都是等距选取的,没有提到指数分割的方法,例如参见非专利文献Khurana,Udayan,and Amol Deshpande."Efficient snapshot retrieval over historical graph data."Data Engineering(ICDE),2013 IEEE 29th International Conference on.IEEE,2013.中的介绍。使用平衡函数的DeltaGraph需要 $O(N \log N)$ 的空间,其中N为到目前为止全部的事件个数,而本发明实施例的方法只需 $O(N)$ 的空间开销,比DeltaGraph更优。获取时序图的快照时,无论查询的时间点在什么位置,使用平衡函数的DeltaGraph的时间为 $O(N)$ ,而本发明实施例的方法的时间复杂度为 $O(m)$ ,其中m为待查询的时间点处时序图快照中事件的个数。对于一个正常增长的时序图来说,如果查询时间点比较靠前,往往会有 $m \ll N$ 。因此,本发明实施例的方法在时间开销上也更优。

[0094] 根据本发明另一实施例,除了进行时间分割外,还可以对图进行分割,也就是把图按照顶点进行进一步划分。这是因为,对于真实的应用场景,只做时间分割是不够的。例如,对于一个单增(即,只有添加顶点或边的事件,而没有删除顶点或边的事件)的时序图G,所有事件都只增加顶点和边,而没有删除的事件。当时序图G演化时,快照的大小变得越来越大,这样会导致封印数据块的大小也越来越大。这种情况下,封印操作所需的时间会很长,从而导致整个系统会在封印操作时遇到性能的冲击。

[0095] 在一个示例中,可以考察时序图某时刻的空间-时间数据块的快照,如果快照的大小超过预定阈值时,将空间-时间数据块的顶点集合分割为两个不相交的第一顶点集合和第二顶点集合,进而关于第一顶点集合和第二顶点集合分别形成对应的快照和日志,从而形成各自的空间-时间数据块。

[0096] 在一个示例中,只在做时间分割的同时做图分割,也就是在做时间分割时,进一步



检查新产生的未封印数据块大小是否超过了数据块大小的阈值。如果超过了阈值,则将产生两个新的未封印数据块,每个可以包含大约一半的事件。例如,当快照 $G_t = (V_t, E_t)$ 的大小到达阈值时,可以把 $V_t$ 分成两个不相交的集合 $V_1$ 和 $V_2$ 。然后把所有 $V_1$ 相关的事件组织在一个数据块中,而所有 $V_2$ 相关的事件组织在另一个数据块中。当图的分割完成后,每个新的未封印数据块都将独立生长。这样,分割操作的开销就控制在了阈值内,平摊到整个时序图的生长过程中。图分割操作实际上是把图按照顶点划分成了两个部分。为了优化基于图遍历的查询,邻近的顶点最好能放在同一个数据块中。图划分算法正好满足这一要求。可以使用已有的METIS[60]图划分算法,有关METIS图划分算法的介绍,可参考非专利文件Karypis, George, and Vipin Kumar. "A fast and high quality multilevel scheme for partitioning irregular graphs." SIAM Journal on scientific Computing 20.1 (1998): 359-392.。

[0097] 在某些情况下,例如基于应用的需求,可能需要重新排列空间-时间数据块中顶点的数据段的物理放置顺序。

[0098] 例如对于图数据来说,遍历是一种常用的查询模式。在基于遍历的查询中,首先会访问一个顶点本身的数据,然后是它的边和对应的邻居顶点。例如,二阶邻居查询给定一个顶点 $v^*$ ,要求访问它的邻居顶点和邻居的邻居顶点。本实施例系统的基于时间分割和图分割的设计为改进数据局部性创造了机会,数据存入系统后可以应基于遍历的查询的需要进行优化。由于图数据最终都保存在磁盘上,而磁盘是以块为基本单位进行读写的,因此不同的顶点顺序会影响对基于遍历的查询。如图9所示,对于同样的图结构,不同的顶点顺序可能会导致需要访问的块数量不同。为了遍历这个子图中的顶点 $v_4$ ,假设只有 $v_4$ 、 $v_1$ 和 $v_9$ 需要被访问。对于顺序1而言,这3个顶点的数据在同一个磁盘的块中,因此只需要访问一个块的内容。而对于顺序2,这3个顶点的数据分别在3个不同的块中,同样的查询需要访问3个块的内容。这会导致不一样的查询性能。在产生封印数据块时,可以通过重新排列磁盘上的顶点顺序的方法来改进基于遍历的查询的数据局部性。例如,对于像二阶邻居查询这样的图遍历查询,我们希望相邻顶点的数据能放置在连续的存储地址里。由于计算图在直线上的最优顺序使得相邻顶点尽量靠近是一个NP-hard问题,优选地可以使用像广度优先搜索这样的启发式方法来得到相对较好的结果。从实现的角度来说,由于封印数据块有索引表,重新排序顶点顺序只需计算出新的顶点顺序,然后在生成封印数据块时按照新的顺序放置顶点数据即可。

[0099] 顶点的数据段在磁盘上的存放顺序的确定优选在封印数据块形成时进行,不过也可以在封印数据块已形成后,基于应用的需要,重新确定适于该应用的顶点的数据段的存储顺序,并且按照新的顺序放置顶点数据,然后更新数据块内部索引,使得基于顶点ID能够定位到重新排列后的顶点的数据段。

[0100] 本发明实施例以二维空间-时间数据块 $C = (V_c, T_c)$ 形式组织随时序图,非常便于查询操作。

[0101] 如前所述,查询一般分全局查询和局部查询,全局查询访问时序图在给定时刻 $t$ 的快照中存在的所有顶点和边,局部查询只访问某时刻的某个顶点或边。更广泛的意义上,查询还可以是处于局部查询和全局查询之间级别的查询,例如查询时序图中某个时刻的特定的某些顶点,或者某个时间区间的某个顶点,某个时间区间的某些顶点,所有这些查询都可

以视为局部查询的组合。

[0102] 下面结合图10描述根据本发明实施例的时序图的特定时刻 $t$ 的全局查询方法10000的过程。

[0103] 如图10所示,在步骤S10100中,基于时刻 $t$ ,基于全局数据块索引获得 $t$ 时刻的所有空间-时间数据块。

[0104] 在步骤S10200中,对于各个空间-时间数据块,扫描空间-时间数据块获得与数据块相关联的查询结果,其中扫描一个空间-时间数据块获得与该空间-时间数据块相关联的查询结果包括:顺序地扫描整个空间-时间数据块,跳过发生时刻大于 $t$ 的所有事件;以及对于任意的一个顶点、一条边,或者一个属性,算法只输出在时刻 $t$ 之前或正好发生在时刻 $t$ 的最后一个相关的事件;另外,如果发现最新的一个事件是删除事件,则该对象在时刻 $t$ 已被删除,同样也不需要输出。

[0105] 在步骤S10300中,合并所有空间-时间数据块的查询结果,获得最终查询结果。

[0106] 下面结合图11描述根据本发明实施例的局部查询方法11000的过程。如果对顶点 $v$ 在时刻 $t$ 的局部查询落在空间-时间数据块 $C$ 中,也就是同时满足 $v \in V_c$ 和 $t \in T_c$ ,则查询结果都在数据块 $C$ 的一个数据段内。

[0107] 如图11所示,在步骤S11100中,基于给定的顶点ID和时刻 $t$ ,通过扫描全局数据块索引而定位到相关联的数据块。

[0108] 在步骤S11200中,基于顶点ID,扫描该数据块的数据块内部索引,定位到数据块中与该顶点ID相关联的数据段。

[0109] 在步骤S11300中,对相关联的数据段做一次随机I/O访问。

[0110] 根据本发明实施例的空间-时间数据块的布局使得局部查询的I/O开销很小,在封印的空间-时间数据块的布局的情况下,局部查询的I/O开销最小。

[0111] 类似地,在需要进行的查询涉及给定顶点或给定顶点集合和涉及给定时刻或给定时间区间的情况下,查询过程可以如下:基于该给定顶点或给定顶点集合和给定时刻或给定时间区间,查询全局数据块索引,定位到相关联的空间-时间数据块;对于定位到的相关联的空间-时间数据块的每个,查询与该空间-时间数据块相关联的数据块内部索引,定位到与给定顶点或给定顶点集合相关联的具体数据段;以及扫描该具体数据段,返回该具体数据段的查询结果;以及合并各个查询结果,并返回合并得到的查询结果。

[0112] 下面结合图12描述根据本发明实施例提供的图数据管理装置。图12示出了根据本发明实施例提供的图数据管理装置12000的结构示意图。

[0113] 如图12所示,图数据组织装置12000可以包括:事件摄入部件12100,配置为摄入新发生的事件,并发送至时序图数据管理引擎;查询引擎12200,配置为接收来自外部的查询,并将该查询发送至时序图数据管理引擎,接收来自时序图数据管理引擎的查询结果,并输出该查询结果;以及时序图数据管理引擎12300,以二维空间-时间数据块 $C = (V_c, T_c)$ 形式组织时序图的数据并存储在存储设备上,一个维度是时间维度,另一个维度是顶点维度,数据块 $C = (V_c, T_c)$ 保存一个时间区间 $[s_c, t_c]$ 中与顶点集合 $V_c$ 相关的数据,所述数据块 $C = (V_c, T_c)$ 逻辑上包括在时刻 $s_c$ 处图的快照以及在时间区间 $[s_c, t_c]$ 内发生事件的日志,其中 $V_c$ 是顶点集合, $T_c$ 指示时间区间, $T_c = [s_c, t_c]$ , $s_c$ 表示该时间区间的起始时刻, $t_c$ 表示该时间区间的结束时刻,在时刻 $s_c$ 处图的快照包括在时刻 $s_c$ 处有效的事件的数据的集合。

[0114] 有关事件摄入部件12100、查询引擎12200以及时序图数据管理引擎12300的具体功能及实现可以参考前面对于空间-时间数据块组织时序图的方法和查询方法的描述,这里不再赘述。

[0115] 以上已经描述了本发明的各实施例,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。因此,本发明的保护范围应该以权利要求的保护范围为准。

1000

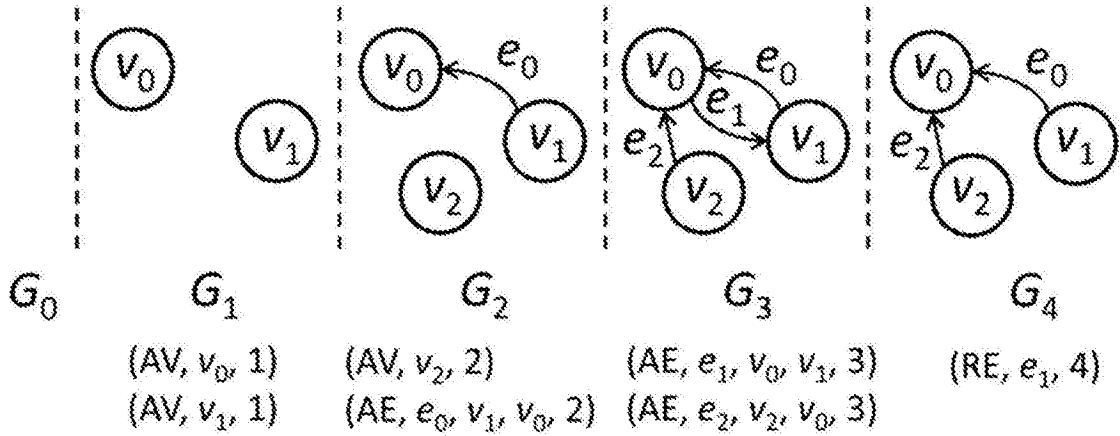


图1

2000

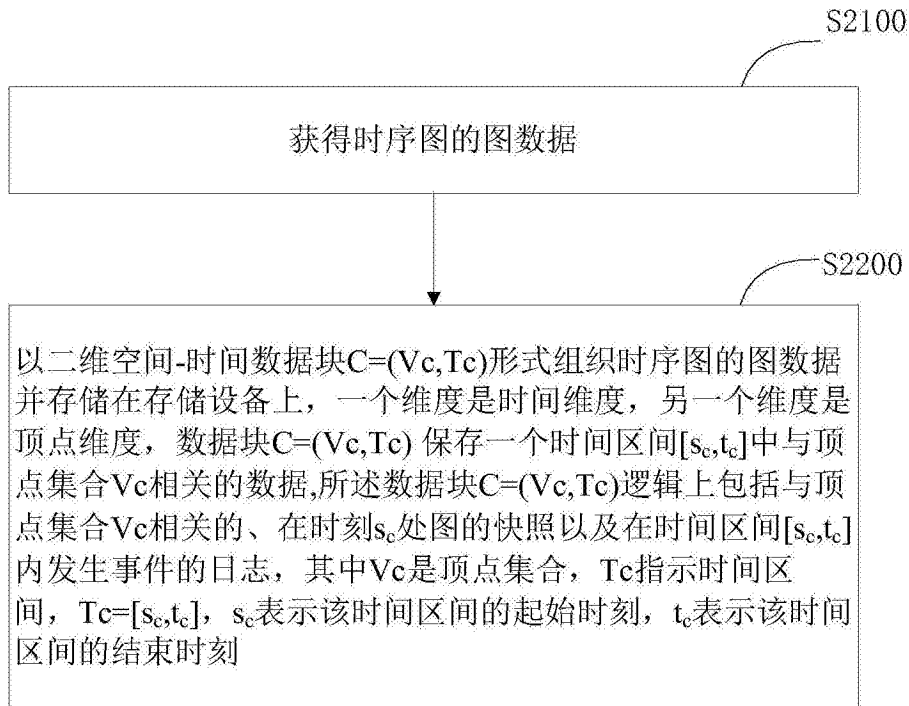


图2

**3000**

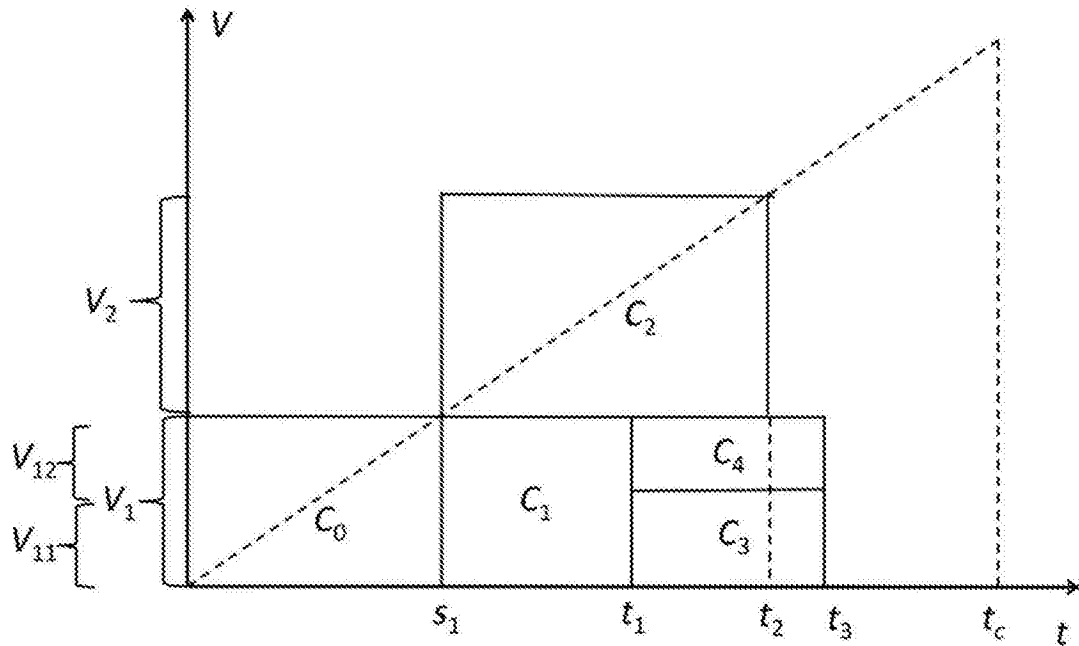


图3

**4000**

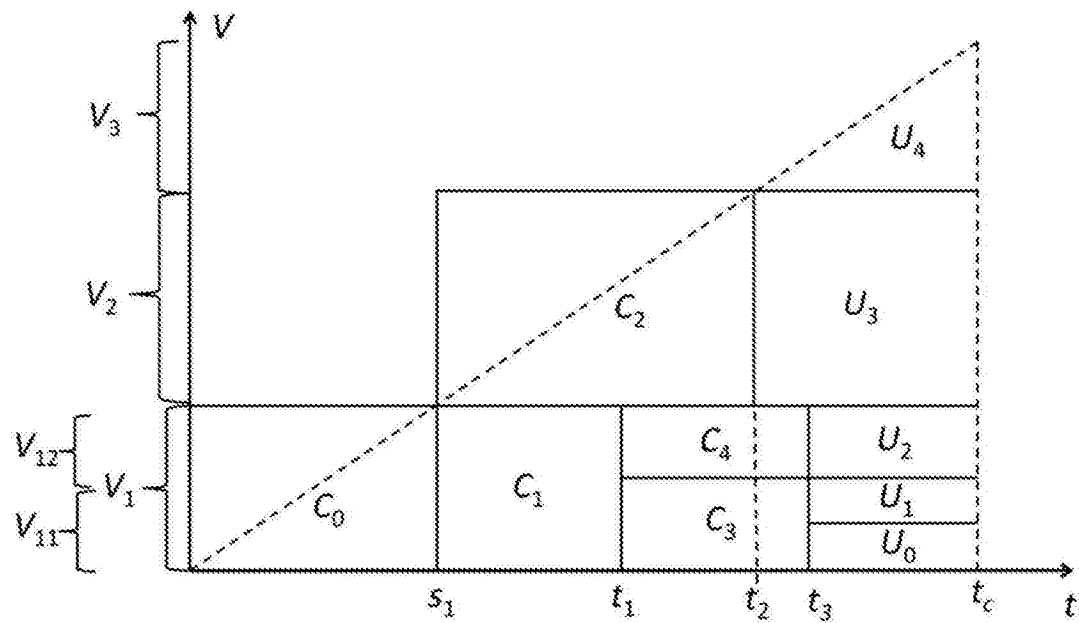


图4

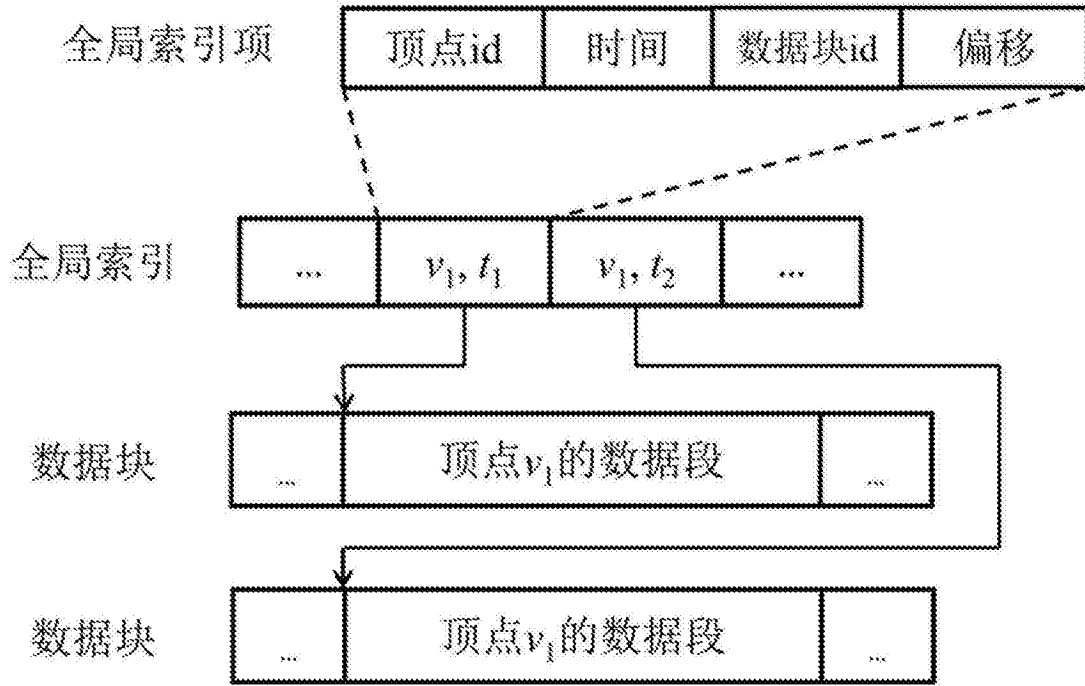


图5

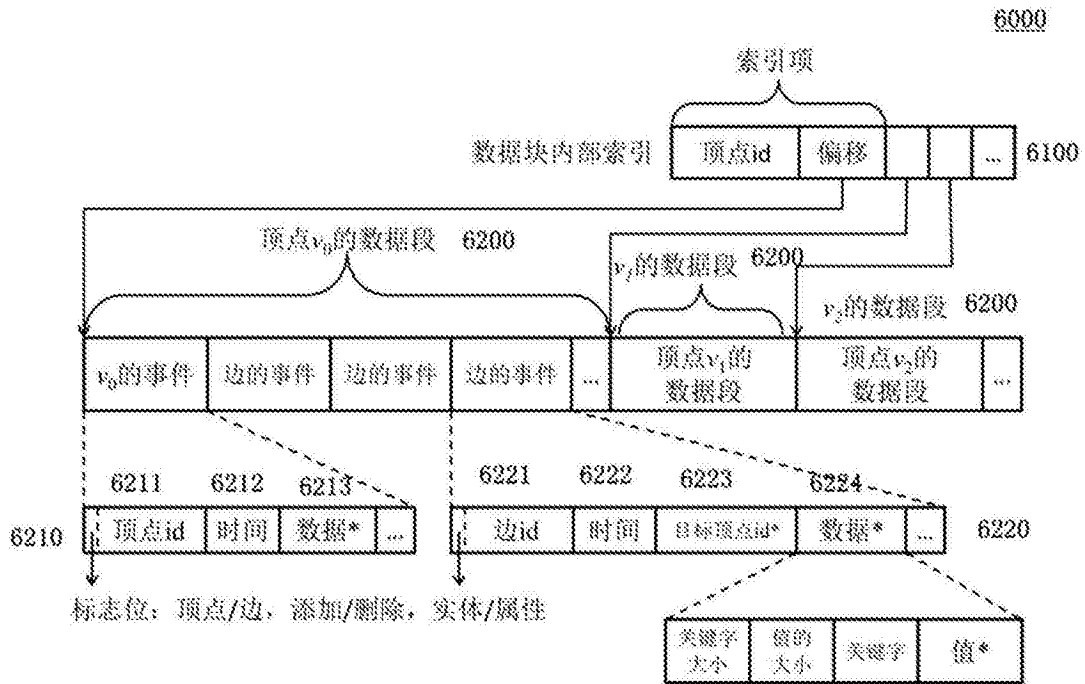


图6

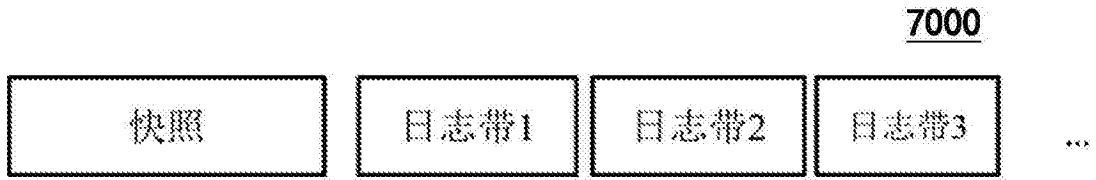


图7

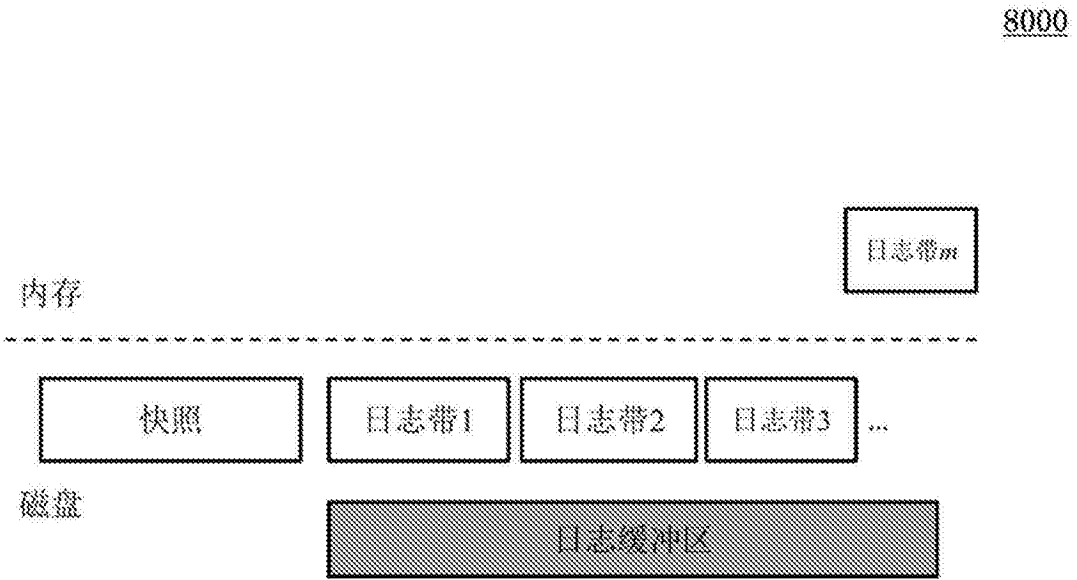


图8

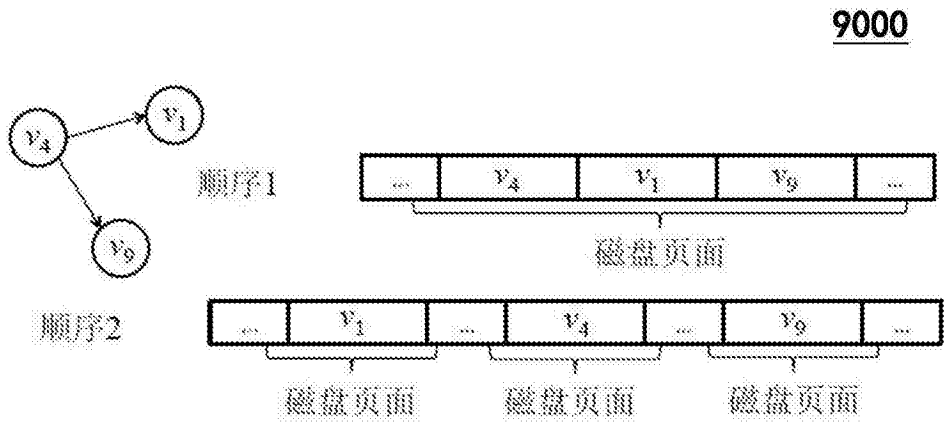


图9

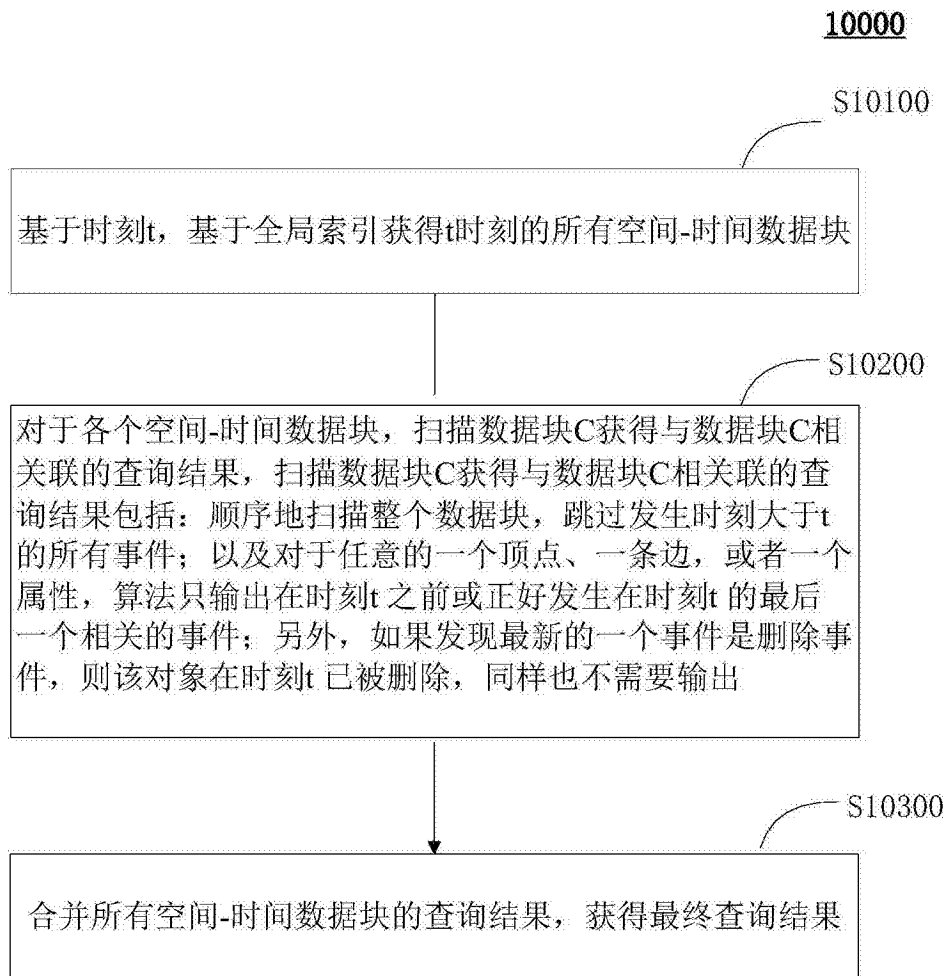


图10



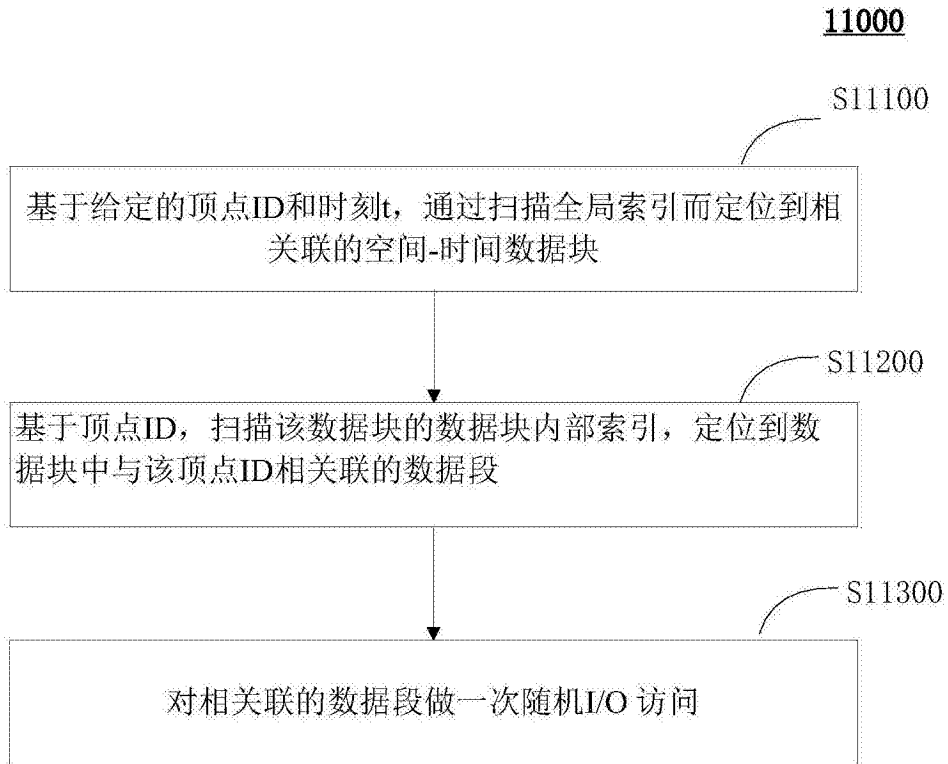


图11

12000

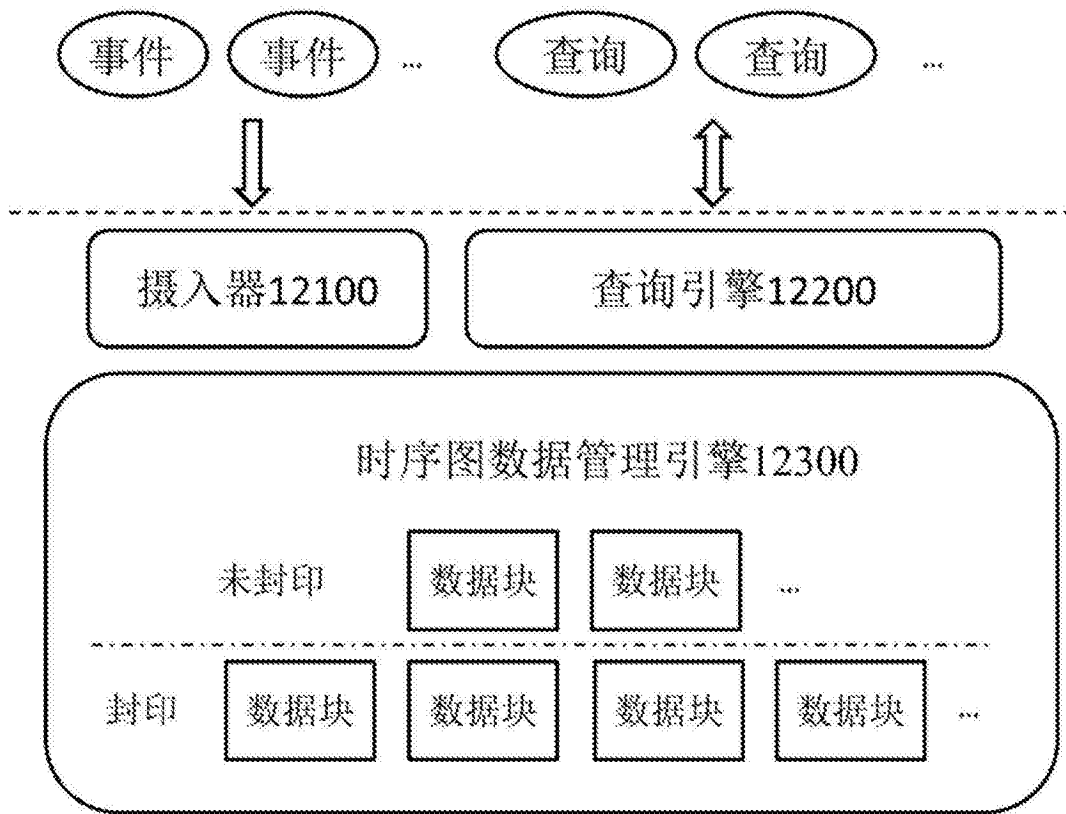


图12