



(12)发明专利申请

(10)申请公布号 CN 106919694 A

(43)申请公布日 2017.07.04

(21)申请号 201710131405.2

(22)申请日 2017.03.07

(71)申请人 郑州云海信息技术有限公司

地址 450018 河南省郑州市郑东新区心怡路278号16层1601室

(72)发明人 吴韶华

(74)专利代理机构 北京集佳知识产权代理有限公司 11227

代理人 罗满

(51)Int.Cl.

G06F 17/30(2006.01)

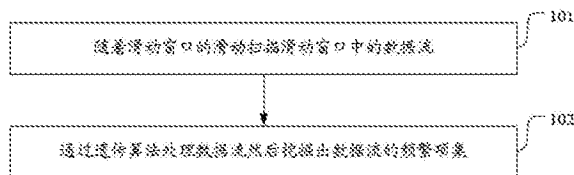
权利要求书2页 说明书7页 附图3页

(54)发明名称

一种基于KNL集群的近期数据流项集挖掘方法与装置

(57)摘要

本发明实施例公开了一种基于KNL集群的近期数据流项集挖掘方法与装置,通过KNL集群的嵌套窗口滑动扫描数据流和遗传算法对数据流进行处理,实现对数据流频繁项集的挖掘,解决了目前在数据流频繁项挖掘中使用模式树造成的空间复杂度过大、用户的操作体验低的技术问题。本发明实施例方法包括:随着滑动窗口的滑动扫描所述滑动窗口中的数据流;通过遗传算法处理所述数据流然后挖掘出所述数据流的频繁项集。



1. 一种基于KNL集群的近期数据流项集挖掘方法,其特征在于,包括:
随着滑动窗口的滑动扫描所述滑动窗口中的数据流;
通过遗传算法处理所述数据流然后挖掘出所述数据流的频繁项集。
2. 根据权利要求1所述的基于KNL集群的近期数据流项集挖掘方法,其特征在于,在扫描滑动窗口中的数据流之前,还包括:
设定滑动窗口的大小为W1。
3. 根据权利要求2所述的基于KNL集群的近期数据流项集挖掘方法,其特征在于,在随着滑动窗口的滑动扫描所述滑动窗口中的数据流之后,在通过遗传算法处理所述数据流然后挖掘出所述数据流的频繁项集之前还包括:
将所述滑动窗口分割为若干个大小为W2的嵌套窗口。
4. 根据权利要求3所述的基于KNL集群的近期数据流项集挖掘方法,其特征在于,通过遗传算法处理所述数据流然后挖掘出所述数据流的频繁项集具体为:
通过并行遗传算法对若干个大小为W2的所述嵌套窗口中的数据流进行第一频繁项集挖掘;
对若干个所述嵌套窗口对应的所述第一频繁项集合并扫描获得所述滑动窗口中所述数据流的第二频繁项集。
5. 根据权利要求4所述的基于KNL集群的近期数据流项集挖掘方法,其特征在于,在对若干个所述嵌套窗口对应的所述第一频繁项集合并扫描获得所述滑动窗口中所述数据流的第二频繁项集之后,还包括:
扫描数据流流动后的滑动窗口的新数据流,然后挖掘数据流流动后的嵌套子窗口的第三频繁项集,根据所述第三频繁项集和数据流未流动的嵌套子窗口的第四频繁项集计算数据流流动后的滑动窗口的第五频繁项集。
6. 一种基于KNL集群的近期数据流项集挖掘装置,其特征在于,包括:
扫描单元,用于随着滑动窗口的滑动扫描所述滑动窗口中的数据流;
频繁项集挖掘单元,用于通过遗传算法处理所述数据流然后挖掘出所述数据流的频繁项集。
7. 根据权利要求6所述的近期数据流项集挖掘装置,其特征在于,还包括:
窗口设定单元,用于设定滑动窗口的大小为W1。
8. 根据权利要求7所述的近期数据流项集挖掘装置,其特征在于,还包括:
嵌套窗口确定单元,用于将所述滑动窗口分割为若干个大小为W2的嵌套窗口。
9. 根据权利要求8所述的近期数据流项集挖掘装置,其特征在于,频繁项集挖掘单元包括嵌套窗口频繁项集挖掘子单元和合并子单元:
嵌套窗口频繁项集挖掘子单元,用于通过并行遗传算法对若干个大小为W2的所述嵌套窗口中的数据流进行第一频繁项集挖掘;
合并子单元,用于对若干个所述嵌套窗口对应的所述第一频繁项集合并扫描获得所述滑动窗口中所述数据流的第二频繁项集。
10. 根据权利要求9所述的近期数据流项集挖掘装置,其特征在于,还包括:
频繁项集更新单元,用于扫描数据流流动后的滑动窗口的新数据流,然后挖掘数据流流动后的嵌套子窗口的第三频繁项集,根据所述第三频繁项集和数据流未流动的嵌套子窗

口的第四频繁项集计算数据流流动后的滑动窗口的第五频繁项集。

一种基于KNL集群的近期数据流项集挖掘方法与装置

技术领域

[0001] 本发明涉及高性能计算领域,尤其涉及一种基于KNL集群的近期数据流项集挖掘方法与装置。

背景技术

[0002] 数据流实际上就是连续移动的元素队伍,其中的元素是由相关数据的集合组成。令 t 表示任一时间戳, a_t 表示在该时间戳到达的数据,流数据可以表示成 $\{\dots, a_{t-1}, a_t, a_{t+1}, \dots\}$ 。区别于传统应用模型,流数据模型具有以下4点共性:(1)数据实时到达;(2)数据到达次序独立,不受应用系统所控制;(3)数据规模宏大且不能预知其最大值;(4)数据一经处理,除非特意保存,否则不能被再次取出处理,或者再次提取数据代价昂贵。

[0003] 滑动窗口(sliding window)模型:滑动窗口对窗口起点和终点都没有明确给定,只明确给定窗口的长度 W_1 。窗口保持一定长度在数据流 $D = \{d_0, d_1, \dots, d_n\}$ 上滑动,处理的数据流范围就由该窗口确定,随着窗口的滑动不断地把得到的结果输出。滑动窗口SW的长度既可由一个时间区间确定,也可由窗口所包含数据流元素个数确定;

[0004] 滑动窗口的频繁项集:对于当前滑动窗口内数据,设 $I = \{i_1, i_2, \dots, i_n\}$ 是项的集合,事务数据集 $S = \{s_0, s_1, \dots, s_n\}$,其中,数据集中每个事务 s 是项的集合, $s \subseteq I$ 。如果 $X \subseteq s$,则称 X 是个项集。如果 X 中有 k 个元素,则称 X 为 k -项集。对于一个项集 X ,如果其支持度大于等于用户给定的最小支持度阈值,则 X 为频繁项集。

[0005] 目前在数据流频繁项挖掘中使用的方法是模式树,会造成空间复杂度过大、用户的操作体验低的技术问题。

发明内容

[0006] 本发明实施例提供了一种基于KNL集群的近期数据流项集挖掘方法与装置,通过KNL集群的嵌套窗口滑动扫描数据流和遗传算法对数据流进行处理,实现对数据流频繁项集的挖掘,解决了目前在数据流频繁项挖掘中使用模式树造成的空间复杂度过大、用户的操作体验低的技术问题。

[0007] 本发明实施例提供了一种基于KNL集群的近期数据流项集挖掘方法,包括:

[0008] 随着滑动窗口的滑动扫描所述滑动窗口中的数据流;

[0009] 通过遗传算法处理所述数据流然后挖掘出所述数据流的频繁项集。

[0010] 优选地,

[0011] 在扫描滑动窗口中的数据流之前,还包括:

[0012] 设定滑动窗口的大小为 W_1 。

[0013] 优选地,

[0014] 在随着滑动窗口的滑动扫描所述滑动窗口中的数据流之后,在通过遗传算法处理所述数据流然后挖掘出所述数据流的频繁项集之前还包括:

[0015] 将所述滑动窗口分割为若干个大小为 W_2 的嵌套窗口。

- [0016] 优选地，
- [0017] 通过遗传算法处理所述数据流然后挖掘出所述数据流的频繁项集具体为：
- [0018] 通过并行遗传算法对若干个大小为 W_2 的所述嵌套窗口中的数据流进行第一频繁项集挖掘；
- [0019] 对若干个所述嵌套窗口对应的所述第一频繁项集合并扫描获得所述滑动窗口中所述数据流的第二频繁项集。
- [0020] 优选地，
- [0021] 在对若干个所述嵌套窗口对应的所述第一频繁项集合并扫描获得所述滑动窗口中所述数据流的第二频繁项集之后，还包括：
- [0022] 扫描数据流流动后的滑动窗口的新数据流，然后挖掘数据流流动后的嵌套子窗口的第三频繁项集，根据所述第三频繁项集和数据流未流动的嵌套子窗口的第四频繁项集计算数据流流动后的滑动窗口的第五频繁项集。
- [0023] 本发明实施例提供了一种基于KNL集群的近期数据流项集挖掘装置，包括：
- [0024] 扫描单元，用于随着滑动窗口的滑动扫描所述滑动窗口中的数据流；
- [0025] 频繁项集挖掘单元，用于通过遗传算法处理所述数据流然后挖掘出所述数据流的频繁项集。
- [0026] 优选地，
- [0027] 所述近期数据流项集挖掘装置还包括：
- [0028] 窗口设定单元，用于设定滑动窗口的大小为 W_1 。
- [0029] 优选地，
- [0030] 所述近期数据流项集挖掘装置还包括：
- [0031] 嵌套窗口确定单元，用于将所述滑动窗口分割为若干个大小为 W_2 的嵌套窗口。
- [0032] 优选地，
- [0033] 频繁项集挖掘单元包括嵌套窗口频繁项集挖掘子单元和合并子单元：
- [0034] 嵌套窗口频繁项集挖掘子单元，用于通过并行遗传算法对若干个大小为 W_2 的所述嵌套窗口中的数据流进行第一频繁项集挖掘；
- [0035] 合并子单元，用于对若干个所述嵌套窗口对应的所述第一频繁项集合并扫描获得所述滑动窗口中所述数据流的第二频繁项集。
- [0036] 优选地，
- [0037] 所述近期数据流项集挖掘装置还包括：
- [0038] 频繁项集更新单元，用于扫描数据流流动后的滑动窗口的的新数据流，然后挖掘数据流流动后的嵌套子窗口的第三频繁项集，根据所述第三频繁项集和数据流未流动的嵌套子窗口的第四频繁项集计算数据流流动后的滑动窗口的第五频繁项集。
- [0039] 从以上技术方案可以看出，本发明实施例具有以下优点：
- [0040] 本发明实施例提供了一种基于KNL集群的近期数据流项集挖掘方法与装置，通过KNL集群的嵌套窗口滑动扫描数据流和遗传算法对数据流进行处理，实现对数据流频繁项集的挖掘，解决了目前在数据流频繁项挖掘中使用模式树造成的空间复杂度过大、用户的操作体验低的技术问题，同时还解决当前服务器计算系统计算网络带宽不足、内存带宽和容量小等情况，能够对较大规模数据进行处理。

附图说明

[0041] 为了更清楚地说明本发明实施例或现有技术中的技术方案,下面将对实施例或现有技术描述中所需要使用的附图作简单地介绍,显而易见地,下面描述中的附图仅仅是本发明的一些实施例,对于本领域普通技术人员来讲,在不付出创造性劳动性的前提下,还可以根据这些附图获得其它的附图。

[0042] 图1为本发明实施例提供的一种基于KNL集群的近期数据流项集挖掘方法的第一实施例的流程示意图;

[0043] 图2为本发明实施例提供的一种基于KNL集群的近期数据流项集挖掘方法的第二实施例的流程示意图;

[0044] 图3为本发明实施例提供的一种基于KNL集群的近期数据流项集挖掘方法的第一实施例的结构示意图;

[0045] 图4为本发明实施例提供的一种基于KNL集群的近期数据流项集挖掘方法的第二实施例的结构示意图;

[0046] 图5为遗传算法流程示意图。

具体实施方式

[0047] 本发明实施例提供了一种基于KNL集群的近期数据流项集挖掘方法与装置,通过KNL集群的嵌套窗口滑动扫描数据流和遗传算法对数据流进行处理,实现对数据流频繁项集的挖掘,解决了目前在数据流频繁项挖掘中使用模式树造成的空间复杂度过大、用户的操作体验低的技术问题。

[0048] 为使得本发明的发明目的、特征、优点能够更加的明显和易懂,下面将结合本发明实施例中的附图,对本发明实施例中的技术方案进行清楚、完整地描述,显然,下面所描述的实施例仅仅是本发明一部分实施例,而非全部的实施例。基于本发明中的实施例,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,都属于本发明保护的范围。

[0049] 本发明实施例提供的一种基于KNL集群的近期数据流项集挖掘方法与装置是基于KNL集群实现的,为了便于对本发明的理解,

[0050] 首先对KNL集群进行说明:KNL集群包括由一台CPU主服务器控制的多台KNL以及相应的存储系统,存储系统的节点、KNL的节点以及CPU主服务器的节点通过高速网络连接,从而实现各节点间的高速通信;其中,KNL是Intel第二代MIC架构Xeon Phi融合处理器Knights Landing的缩写,是Intel首款专门针对高度并行工作负载而设计的可独立自启动的主处理器,并首次实现了内存与高速互连技术的集成,KNL提供能够独立自启动的Bootable形态,从而消除对PCI-E总线的依赖性,实现更高效扩展;每个KNL包含72个核心,KNL平台每个节点包含16GMCDRAM内存,其传输速度可以达到400G/s以上,最大支持384G的DDR4内存,相比DDR4的90G/s有非常大的提升,每个核心支持四个线程,KNL处理器实现了独立作为主处理器的功能,其在KNC的基础上实现了极大的提升,本发明在充分利用KNL的硬件的基础上,充分利用MCADRAM的内存空间,将图片的关键点信息和关键点的梯度等信息开辟内存到MCDRAM上,以便在特征提取过程中将减少访存cache hit的次数,达到快速访存的

目的,实现了数据挖掘的KNL众核集群平台的加速;CPU主服务器的控制节点控制KNL的计算节点,并对计算节点进行统一调度,实现数据的统一分配,同时实现通信传输与并行计算的异步执行,提高了整体的运行效率,满足了高性能应用的要求;对于KNL,两个KNL处理器的核心共享一个1M的L2缓存,本发明实施例尽量在loop中使得访存可以控制在L2缓存大小以内,可以快速的完成循环,加快特征提取过程中提取速度,且将每个loop中的访存地址区块化,避免跳跃式访存,实现程序的代码级别优化;在KNL集群中,为了充分利用KNL的多线程技术,开启openmp多线程并行,从而在最大限度上利用KNL计算能力;另一方面在程序运行过程中通过采用分块技术,使得二级缓存可以得到充分的利用,减少由于访存带来的延迟。

[0051] 然后,对数据流进行说明:数据流实际上就是连续移动的元素队伍,其中的元素是由相关数据的集合组成;令 t 表示任一时间戳, a_t 表示在该时间戳到达的数据,流数据可以表示成 $\{\dots, a_{t-1}, a_t, a_{t+1}, \dots\}$;区别于传统应用模型,流数据具有以下4点共性:(1)数据实时到达;(2)数据到达次序独立,不受应用系统所控制;(3)数据规模宏大且不能预知其最大值;(4)数据一经处理,除非特意保存,否则不能被再次取出处理,或者再次提取数据代价昂贵;

[0052] 其次,对滑动窗口进行说明:滑动窗口对窗口起点和终点都没有明确给定,只明确给定窗口的长度 W_1 ;窗口保持一定长度在数据流 $D = \{d_0, d_1, \dots, d_n\}$ 上滑动,处理的数据流范围就由该窗口确定,随着窗口的滑动不断地把得到的结果输出;滑动窗口 SW 的长度既可由一个时间区间确定,也可由窗口所包含数据流元素个数确定;

[0053] 然后再对滑动窗口的频繁项集进行说明:对于当前滑动窗口内数据,设 $I = \{i_1, i_2, \dots, i_n\}$ 是项的集合,事务数据集 $S = \{s_0, s_1, \dots, s_n\}$,其中,数据集中每个事务 s 是项的集合, $s \subseteq I$;如果 $X \subseteq s$,则称 X 是个项集;如果 X 中有 k 个元素,则称 X 为 k -项集;对于一个项集 X ,如果其支持度大于等于用户给定的最小支持度阈值,则 X 为频繁项集;

[0054] 最后对遗传算法进行说明:遗传算法是一种基于随机搜索的优化算法,已成功应用于函数优化、自动控制、生产调度、机器人学、图像处理、人工生命、机器学习和数据挖掘等领域。从代表问题可能潜在的解集的一个种群开始的,而一个种群则由经过基因编码的一定数目的个体组成。每个个体实际上是染色体带有特征的实体。染色体作为遗传物质的主要载体,即多个基因的集合,其内部表现(即基因型)是某种基因组合,它决定了个体的形状的外部表现,如黑头发的特征是由染色体中控制这一特征的某种基因组合决定的。因此,在一开始需要实现从表现型到基因型的映射即编码工作。由于仿照基因编码的工作很复杂,我们往往进行简化,如二进制编码,初代种群产生之后,按照适者生存和优胜劣汰的原理,逐代演化产生出越来越好的近似解,在每一代,根据问题域中个体的适应度大小选择个体,并借助于自然遗传学的遗传算子进行组合交叉和变异,产生出代表新的解集的种群。这个过程将导致种群像自然进化一样的后世代种群比前代更加适应于环境,末代种群中的最优个体经过解码,可以作为问题近似最优解。

[0055] 遗传算法的基本运算过程如下,算法流程示意图如图5所示:

[0056] a) 初始化:设置进化代数计数器 $t=0$,设置最大进化代数 T ,随机生成 M 个个体作为初始群体 $P(0)$ 。

[0057] b) 个体评价:计算群体 $P(t)$ 中各个个体的适应度。

[0058] c) 选择运算:将选择算子作用于群体。选择的目的是把优化的个体直接遗传到下

一代或通过配对交叉产生新的个体再遗传到下一代。选择操作是建立在群体中个体的适应度评估基础上的。

[0059] d) 交叉运算:将交叉算子作用于群体。所谓交叉是指把两个父代个体的部分结构加以替换重组而生成新个体的操作。遗传算法中起核心作用的就是交叉算子。

[0060] e) 变异运算:将变异算子作用于群体。即是对群体中的个体串的某些基因座上的基因值作变动。群体 $P(t)$ 经过选择、交叉、变异运算之后得到下一代群体 $P(t+1)$ 。

[0061] f) 终止条件判断:若 $t=T$,则以进化过程中所得到的具有最大适应度个体作为最优解输出,终止计算。

[0062] 请参阅图1,本发明实施例提供了一种基于KNL集群的近期数据流项集挖掘方法的第一实施例,包括:

[0063] 101,随着滑动窗口的滑动扫描滑动窗口中的数据流;

[0064] 在本发明实施例中,首先需要随着滑动窗口的滑动扫描滑动窗口中的数据流。

[0065] 102,通过遗传算法处理数据流然后挖掘出数据流的频繁项集;

[0066] 在本发明实施例中,在随着滑动窗口的滑动扫描滑动窗口中的数据流之后,还需要通过遗传算法处理数据流然后挖掘出数据流的频繁项集。

[0067] 请参阅图2,本发明实施例提供了一种基于KNL集群的近期数据流项集挖掘方法的第二实施例,包括:

[0068] 201,设定滑动窗口的大小为 W_1 ;

[0069] 在本发明实施例中,首先需要设定滑动窗口的大小为 W_1 ,滑动窗口的大小 W_1 是由数据流的属性决定的。

[0070] 202,随着滑动窗口的滑动扫描滑动窗口中的数据流;

[0071] 在本发明实施例中,在设定滑动窗口的大小为 W_1 之后,还需要随着滑动窗口的滑动扫描滑动窗口中的数据流。

[0072] 203,将滑动窗口分割为若干个大小为 W_2 的嵌套窗口;

[0073] 在本发明实施例中,在随着滑动窗口的滑动扫描滑动窗口中的数据流之后,还需要将滑动窗口分割为若干个大小为 W_2 的嵌套窗口,其中嵌套窗口的大小 W_2 也是由数据流的属性决定的。

[0074] 204,通过并行遗传算法对若干个大小为 W_2 的嵌套窗口中的数据流进行第一频繁项集挖掘;

[0075] 在本发明实施例中,在将滑动窗口分割为若干个大小为 W_2 的嵌套窗口之后,还需要通过并行遗传算法对若干个大小为 W_2 的嵌套窗口中的数据流进行第一频繁项集挖掘,其中若干个嵌套窗口的第一频繁项集挖掘是通过KNL的若干个节点并行实现的;需要说明的是,通过并行遗传算法进行第一频繁项集挖掘时,首先要设定一个支持度阈值 S ,用来挖掘最终的数据流频繁项集,当 F_i 大于等于 S 时,那么事务 i 就是滑动窗口中的数据频繁项集;还要确定最大迭代次数 T ,主要是根据初始化的种群大小和事务中属性多少及其取值范围来决定;在进行遗传算法的迭代计算之前,要初始化种群,因为数据流的动态更新,这样对于嵌套子窗口,获得最新进入的数据,由此得到频繁模式的1-项集,根据编码规则,将频繁1-项集编码成实数串,如果频繁1-项集有非零项,就按原来的位置进行随机组合编码,这样,就一起构成了嵌套子窗口中的初始种群,其中的个体为准备分析确认的频繁模式;迭代计

算首先计算嵌套窗口中个体支持度的值,判定个体支持度值是否大于设定的支持度阈值S,如果大于设定的支持度阈值S,就将此个体的频繁模式添加到当前嵌套子窗口的频繁项集内,其中实际每一个事务与初始种群中等待分析的频繁模式是通过公式 $F_i = W_i / W_z$ 计算匹配的;然后根据计算出的嵌套窗口中个体支持度的值做轮盘选择操作,再根据预先设定的交叉概率P和变异概率Q对个体进行交叉造作和基本位变异操作;上述操作是一次遗传算子操作,每进行一次遗传算子操作之后,需要将个体支持度值大于等于S的个体,添加到嵌套子窗口的频繁项集中,到此,完成了遗传算法的一次迭代,本发明实施例需要进行若干次迭代,直到迭代次数大于等于预置的最大迭代次数T迭代停止并获得当前嵌套子窗口内数据的稳定频繁项集,否则要从计算嵌套窗口中个体支持度的值开始继续重新进行下一次迭代计算。

[0076] 205,对若干个嵌套窗口对应的第一频繁项集合并扫描获得滑动窗口中数据流的第二频繁项集;

[0077] 在本发明实施例中,在通过并行遗传算法对若干个大小为W2的嵌套窗口中的数据流进行第一频繁项集挖掘之后,还需要对若干个嵌套窗口对应的第一频繁项集合并扫描获得滑动窗口中数据流的第二频繁项集,并通过判断第二频繁项集的支持度大于预置的支持度阈值S确定第二频繁项集为最终频繁项集。

[0078] 206,扫描数据流流动后的滑动窗口的新数据流,然后挖掘数据流流动后的嵌套子窗口的第三频繁项集,根据第三频繁项集和数据流未流动的嵌套子窗口的第四频繁项集计算数据流流动后的滑动窗口的第五频繁项集;

[0079] 在本发明实施例中,在对若干个嵌套窗口对应的第一频繁项集合并扫描获得滑动窗口中数据流的第二频繁项集之后,还需要扫描数据流流动后的滑动窗口的的新数据流,然后挖掘数据流流动后的嵌套子窗口的第三频繁项集,根据第三频繁项集和数据流未流动的嵌套子窗口的第四频繁项集计算数据流流动后的滑动窗口的第五频繁项集,需要说明的是,第四频繁项集是包含在第一频繁项集中的,事实是数据流未流动的嵌套子窗口的第一频繁项集,为了以防混淆,特称其为第一频繁项集。

[0080] 请参阅图3,本发明实施例提供了一种基于KNL集群的近期数据流项集挖掘装置的第一实施例,包括:

[0081] 扫描单元301,用于随着滑动窗口的滑动扫描滑动窗口中的数据流;

[0082] 频繁项集挖掘单元302,用于通过遗传算法处理数据流然后挖掘出数据流的频繁项集。

[0083] 请参阅图4,本发明实施例提供了一种基于KNL集群的近期数据流项集挖掘装置的第二实施例,包括:

[0084] 窗口设定单元401,用于设定滑动窗口的大小为W1。

[0085] 扫描单元402,用于随着滑动窗口的滑动扫描滑动窗口中的数据流;

[0086] 嵌套窗口确定单元403,用于将滑动窗口分割为若干个大小为W2的嵌套窗口。

[0087] 频繁项集挖掘单元404,用于通过遗传算法处理数据流然后挖掘出数据流的频繁项集。

[0088] 频繁项集挖掘单元404包括嵌套窗口频繁项集挖掘子单元4041和合并子单元4042:

[0089] 嵌套窗口频繁项集挖掘子单元4041,用于通过并行遗传算法对若干个大小为W2的嵌套窗口中的数据流进行第一频繁项集挖掘;

[0090] 合并子单元4042,用于对若干个嵌套窗口对应的第一频繁项集合并扫描获得滑动窗口中数据流的第二频繁项集。

[0091] 频繁项集更新单元405,用于扫描数据流流动后的滑动窗口的新数据流,然后挖掘数据流流动后的嵌套子窗口的第三频繁项集,根据第三频繁项集和数据流未流动的嵌套子窗口的第四频繁项集计算数据流流动后的滑动窗口的第五频繁项集。

[0092] 所属领域的技术人员可以清楚地了解到,为描述的方便和简洁,上述描述的系统,装置和单元的具体工作过程,可以参考前述方法实施例中的对应过程,在此不再赘述。

[0093] 在本申请所提供的几个实施例中,应该理解到,所揭露的系统,装置和方法,可以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,机械或其它的形式。

[0094] 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。

[0095] 另外,在本发明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,也可以采用软件功能单元的形式实现。

[0096] 所述集成的单元如果以软件功能单元的形式实现并作为独立的产品销售或使用,可以存储在一个计算机可读取存储介质中。基于这样的理解,本发明的技术方案本质上或者说对现有技术做出贡献的部分或者该技术方案的全部或部分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介质中,包括若干指令用以使得一台计算机设备(可以是个人计算机,服务器,或者网络设备等)执行本发明各个实施例所述方法的全部或部分步骤。而前述的存储介质包括:U盘、移动硬盘、只读存储器(ROM,Read-Only Memory)、随机存取存储器(RAM,Random Access Memory)、磁碟或者光盘等各种可以存储程序代码的介质。

[0097] 以上所述,以上实施例仅用以说明本发明的技术方案,而非对其限制;尽管参照前述实施例对本发明进行了详细的说明,本领域的普通技术人员应当理解:其依然可以对前述各实施例所记载的技术方案进行修改,或者对其中部分技术特征进行等同替换;而这些修改或者替换,并不使相应技术方案的本质脱离本发明各实施例技术方案的精神和范围。

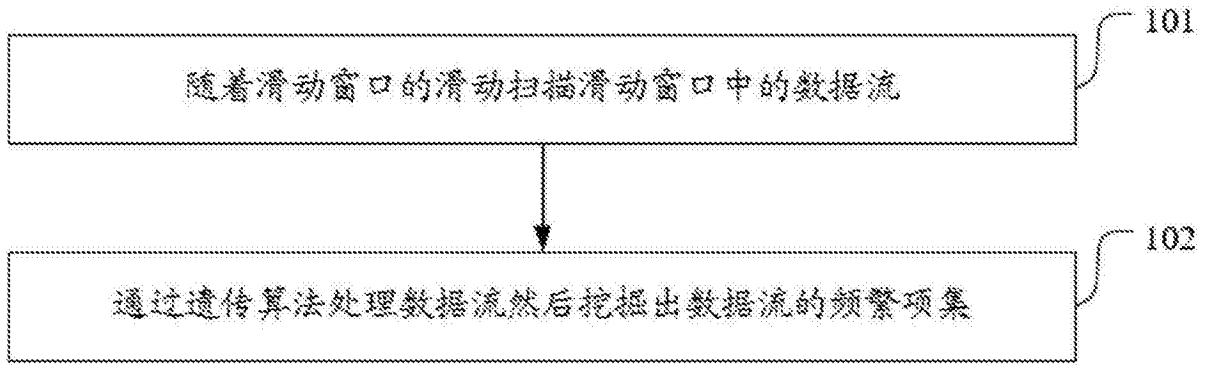


图1

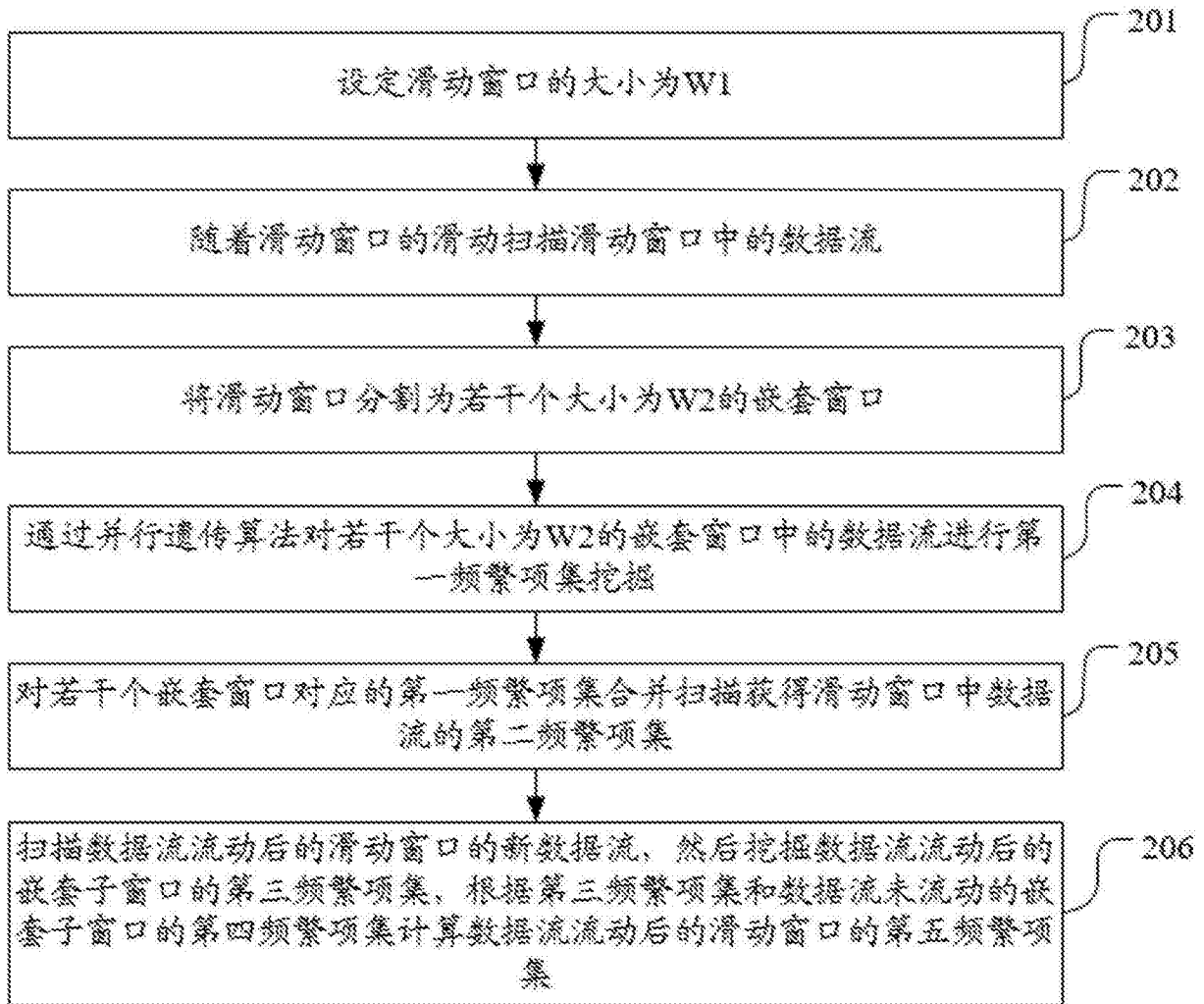


图2

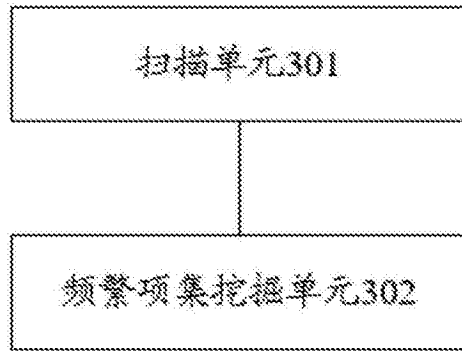


图3

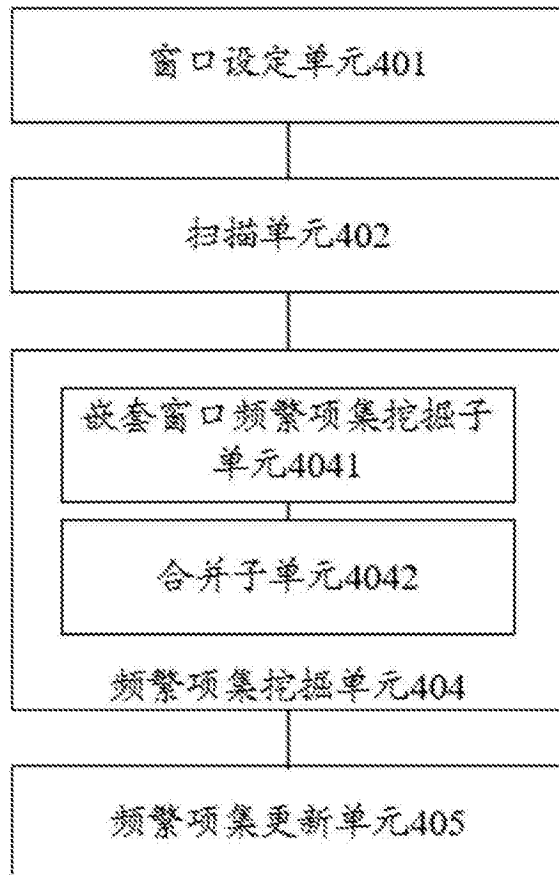


图4

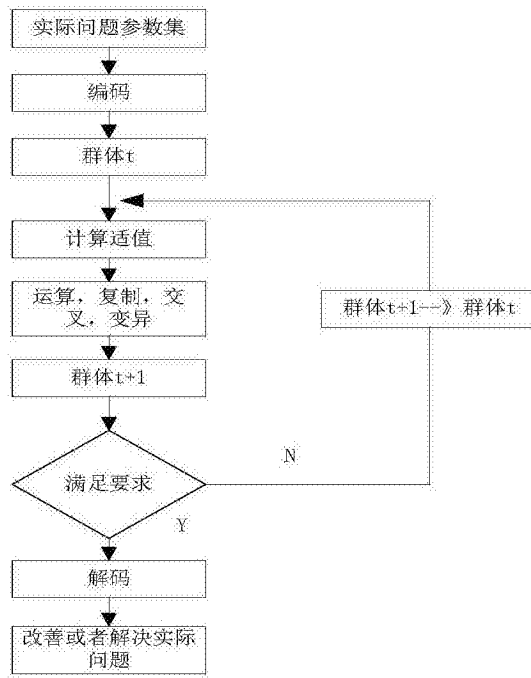


图5