



(12)发明专利

(10)授权公告号 CN 105357247 B

(45)授权公告日 2018.08.28

(21)申请号 201510606941.4

(22)申请日 2015.09.22

(65)同一申请的已公布的文献号

申请公布号 CN 105357247 A

(43)申请公布日 2016.02.24

(73)专利权人 上海理工大学

地址 200093 上海市杨浦区军工路516号

(72)发明人 陈世平 胡凯 蒲云花

(74)专利代理机构 上海申汇专利代理有限公司

31001

代理人 吴宝根

(51)Int.Cl.

H04L 29/08(2006.01)

H04L 29/06(2006.01)

(56)对比文件

CN 101883101 A,2010.11.10,

CN 103139314 A,2013.06.05,

CN 101853283 A,2010.10.06,

CN 101360055 A,2009.02.04,

蒲云花,陈世平,马友礼..基于复杂云资源查询的HChord索引机制.《计算机应用研究》.2014,第31卷(第12期),3818-3821.

张明英,胡德敏,高丽萍,陈世平..基于结构化对等网络的云资源查询算法.《计算机应用研究》.2015,第32卷(第2期),532-535.

李璞,陈世平,李剑锋..一种基于对等网络的云资源定位算法.《计算机应用研究》.2013,第30卷(第2期),570-573.

邵一川,申德荣,赵宏伟..HilbertChord:一种网格资源管理的P2P框架.《微计算机信息》.2009,第25卷(第15期),105-106.

李璞,陈世平..一种基于对等网络的云资源多属性区间查询算法.《计算机应用研究》.2013,第30卷(第9期),2831-2834.

审查员 吴兰花

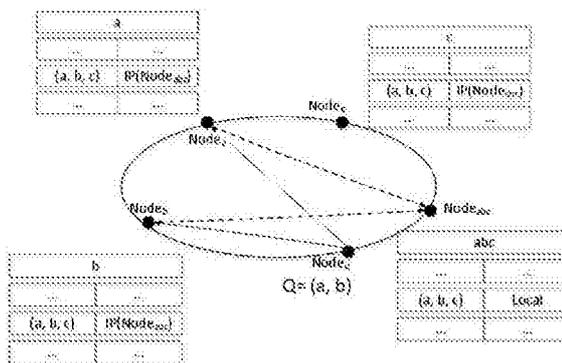
权利要求书2页 说明书6页 附图2页

(54)发明名称

基于分层云对等网络的多维属性云资源区间查找方法

(57)摘要

本发明涉及一种基于分层云对等网络的多维属性云资源区间查找方法,以分层云对等网络为基础,分别利用云资源的资源类型和资源属性值建立多维索引,将相关性的数据聚集存储在一个资源簇内;并将属性值的值域划分为多个区段,以满足更复杂的查询。同时建立资源簇融合、区间邻居维持等机制使该算法能在计算复杂度为对数级的基础上完成检索。充分结合现存结构化P2P网络优点,利用分层结构,将云资源分簇存储,利用资源簇定位技术,以较小的代价快速定位资源簇,同时在资源簇内建立HChord环,保持数据相关性;结构简单,易于实现。具有很好的时间、空间和通信复杂度,查询在对数级跳数内完成,且算法路由跳数不随网络规模的增大而迅速增加。



CN 105357247 B

1. 一种基于分层云对等网络的多维属性云资源区间查找方法,其特征在于,节点N收到包含类型和属性值信息的查询请求QA,如果节点N不是资源簇代理节点D,则将查询请求QA的类型信息发送给D,用节点D解析QA:首先与本节点D提供的资源类型比较,如果匹配,执行资源簇代理节点D内资源簇查找;不匹配则采用资源簇定位算法,然后将查询请求QA同时发送给所有满足该类型的资源簇代理节点,执行资源簇内查找,当资源簇定位算法也找不到代理节点时,向用户反馈查找失败;

其中资源簇内查找:首先依据QA的类型及属性值信息查询区间划分表,确定查询资源的笛卡尔坐标,并转化为HSFC编码;然后依据Chord规则定位到提供资源的节点;最后将资源及节点的IP发送给用户,用户可以直接与该节点通信;如果资源节点提供的资源不足或者无此资源,资源节点将区间邻居节点反馈给用户并附上邻居节点标记,如果区间邻居节点也没有资源,那么再提供邻居的区间邻居节点的资源,再无资源后向用户反馈查找失败;

用户集合资源节点提供的资源信息,当资源来自多个资源簇时,按匹配权重排序,提供给用户。

2. 根据权利要求1所述基于分层云对等网络的多维属性云资源区间查找方法,其特征在于,所述资源簇定位算法:

算法默认所有的网络资源包含资源类型和资源属性值两种属性,并采用双层Chord模型,第一层为实现资源类型信息的索引,第二层为将相关类型的资源聚集在一起而形成的资源簇;节点的加入与离开遵循Chord的规则,

其中索引建立方法:使用资源类型作为索引,根据资源类型的索引向量,利用SHA-1散列函数求哈希值,然后利用Chord规则根据上述所求得到的哈希值确定相应节点,确定的节点内包含了存储资源簇的索引项,就是资源簇的入口节点,在资源簇内再建立资源属性值索引,便定位资源节点。

3. 根据权利要求2所述基于分层云对等网络的多维属性云资源区间查找方法,其特征在于,所述资源簇资源簇内每一类型就是资源空间的一个维度,k维类型就形成一个k维笛卡尔坐标系,每个类型的属性值以其值域被划分为d个区间,这样便形成了 d^k 个多维区间,每个多维区间的属性值向量都可以用这个多维区间的笛卡尔坐标值 $D(x_1, x_2, \dots, x_k)$ 来表示,由于类型个数和区间划分情况都是变动的,因此每个资源簇保存一张区间划分表,依据区间划分表,属性值向量能转化为相应的笛卡尔坐标。

4. 根据权利要求1所述基于分层云对等网络的多维属性云资源区间查找方法,其特征在于,所述匹配权重具体的公式如下:

假设某资源簇的类型维度为C,查询语句的类型维度为 C_q ,则类型的匹配权重为:

$$W_1 = \frac{C_q}{C}$$

假设资源簇内的某资源节点的邻居跳数为X,当资源来自节点本身,X为0;当资源来自直接邻居,X为1,以此类推;则属性值的匹配权重为:

$$W_2 = \frac{1}{1+X}$$

最终的匹配权重为:

$$W = \alpha W_1 + \beta W_2$$

其中, α 为类型匹配因子, β 为属性值匹配因子,可由人工配置;当 w 越大,匹配程度越高,而当资源簇的类型维度与查询的类型维度相差越大,或者查询结果来源于原资源节点的邻居节点,则 w 越小,匹配程度也就越低。

基于分层云对等网络的多维属性云资源区间查找方法

技术领域

[0001] 本发明涉及一种计算机网络技术,特别涉及一种基于分层云对等网络的多维属性云资源区间查找方法。

背景技术

[0002] 在对等(P2P)系统中,数据可存储在网络中的任意结点上。因此,如何能够高效地定位存储特定数据的结点成为一个最基本的问题。而目前云平台网络数据种类多样,并依据多级索引结构化存储,因此单关键字查询已不能满足云系统,而能够快速支持复杂查询已是目前云对等网络查询算法的关键。

[0003] 目前结构化P2P 上多属性查询的研究主要有三种方案:

[0004] 1) 复用单维属性查询来实现多属性查找机制,如BATON、Mercury都是采用这种方式进行多属性查询处理。其最终查询结果是由每一单属性查找结果做交集操作得来的。即便算法有所优化,但查询性能也会随着属性维度的增加而迅速降低。

[0005] 2) 将多维属性资源利用降维曲线映射到单维索引空间。如HilbertChord,其基本原理是用Hilbert曲线穿过多维空间而形成一维索引。而Znet是基于Skip Graphs的覆盖网络,用Z曲线将多属性资源空间映射成一维,以形成支持多关键字查询、多区间查询的索引平台。

[0006] 3) 是直接建立多维索引。如Dak提出一种基于kd-tree索引划分,并将数据索引树内嵌到DHT网络底层,以支持多属性区间查询的索引框架。GChord改造传统的多维索引结构以适应P2P网络环境,并构造一个多播树来实现多维属性的查找。

发明内容

[0007] 本发明是针对进一步改进结构化P2P 上多属性查询的问题,提出了一种基于分层云对等网络的多维属性云资源区间查找方法,在现有技术的优点基础上,进一步改进,本发明能够布局在P2P网络上的多维属性区间查找算法,满足更复杂的查询,算法不会随着网络节点数及资源属性数增加而产生较大查询延迟。

[0008] 本发明的技术方案为:一种基于分层云对等网络的多维属性云资源区间查找方法,节点N收到包含类型和属性值信息的查询请求QA,如果节点N不是资源簇代理节点D,则将查询请求QA的类型信息发送给D,用节点D解析QA:首先与本节点D提供的资源类型比较,如果匹配,执行资源簇代理节点D内资源簇查找;不匹配则采用资源簇定位算法,然后将查询请求QA同时发送给所有满足该类型的资源簇代理节点,执行资源簇内查找,当资源簇定位算法也找不到代理节点时,向用户反馈查找失败;

[0009] 其中资源簇内查找:首先依据QA的类型及属性值信息查询区间划分表,确定查询资源的笛卡尔坐标,并转化为HSFC编码;然后依据Chord规则定位到提供资源的节点;最后将资源及节点的IP发送给用户,用户可以直接与该节点通信;

[0010] 如果资源节点提供的资源不足或者无此资源,资源节点将区间邻居节点反馈给用

户并附上邻居节点标记,如果区间邻居节点也没有资源,那么再提供邻居的区间邻居节点的资源,再无资源后向用户反馈查找失败;

[0011] 用户集合资源节点提供的资源信息,当资源来自多个资源簇时,按匹配权重排序,提供给用户。

[0012] 所述资源簇定位算法:

[0013] 算法默认所有的网络资源包含资源类型和资源属性值两种属性,并采用双层Chord模型,第一层为实现资源类型信息的索引,第二层为将相关类型的资源聚集在一起而形成的资源簇;节点的加入与离开遵循Chord的规则,

[0014] 其中索引建立方法:使用资源类型作为索引,根据资源类型的索引向量,利用SHA-1散列函数求哈希值,然后利用Chord规则根据1中的哈希值确定相应节点,确定的节点内包含了存储资源簇的索引项,就是资源簇的入口节点,在资源簇内再建立资源属性值索引,便定位资源节点。

[0015] 所述资源簇资源簇内每一类型就是资源空间的一个维度,k维类型就形成一个k维笛卡尔坐标系,每个类型的属性值以其值域被划分为d个区间,这样便形成了 d^k 个多维区间,每个多维区间的属性值向量都可以用这个多维区间的笛卡尔坐标值 $D(x_1, x_2, \dots, x_k)$ 来表示,由于类型个数和区间划分情况都是变动的,因此每个资源簇保存一张区间划分表,依据区间划分表,属性值向量能转化为相应的笛卡尔坐标。

[0016] 所述匹配权重具体的公式如下:

[0017] 假设某资源簇的类型维度为C,查询语句的类型维度为 C_q ,则类型的匹配权重为:

$$[0018] \quad W_1 = \frac{C_q}{C}$$

[0019] 假设资源簇内的某资源节点的邻居跳数为X(当资源来自节点本身,X为0;来自直接邻居,X为1,以此类推),则属性值的匹配权重为:

$$[0020] \quad W_2 = \frac{1}{1+X}$$

[0021] 最终的匹配权重为:

$$[0022] \quad W = \alpha W_1 + \beta W_2$$

[0023] 其中, α 为类型匹配因子, β 为属性值匹配因子,可由人工配置,当 W 越大,匹配程度越高,而当资源簇的类型维度与查询的类型维度相差越大,或者查询结果来源于原资源节点的邻居节点,则 W 越小,匹配程度也就越低。

[0024] 本发明的有益效果在于:本发明基于分层云对等网络的多维属性云资源区间查找方法,具有很好的时间复杂度、空间复杂度和通信复杂度,查询在对数级跳数内完成,且算法路由跳数不随网络规模的增大而迅速增加;本发明充分结合了现存结构化P2P网络优点,利用分层P2P网络结构,将云资源分簇存储,利用资源簇定位技术,利用较小的代价快速定位资源簇,同时在资源簇内建立Hilbert-Chord环,保持数据相关性;本发明结构简单,易于实现。

附图说明

- [0025] 图1为本发明实施例HChord网络的基本模型图；
 [0026] 图2为本发明实施例HChord网络的资源簇索引示意图；
 [0027] 图3为本发明实施例HChord网络资源簇内的资源分布空间及Hilbert环示意图。

具体实施方式

[0028] 本发明基于分层云对等网络的多维属性云资源区间查找算法(HChord)主要思想是:以分层云对等网络为基础,分别利用云资源的资源类型和资源属性值建立多维索引,将相关性的数据聚集存储在一个资源簇内;并将属性值的值域划分为多个区段,以满足更复杂的查询。同时建立资源簇融合、区间邻居维持等机制使该算法能在计算复杂度为对数级的基础上完成检索。

[0029] 具体的查询算法步骤如下:

[0030] 1) 节点N收到查询请求QA(QA包含类型和属性值信息),如果节点N不是资源簇代理节点D,则将查询请求QA的类型信息发送给D。用节点D解析QA:首先与本节点D提供的资源类型比较,如果匹配,执行资源簇代理节点D内资源簇查找;不匹配则采用资源簇定位算法,然后将查询请求QA发送给满足该类型的资源簇代理节点(可能不止一个,当有多个时,查找是并行的),以执行资源簇内查找。而当资源簇定位算法也找不到代理节点时,向用户反馈查找失败。

[0031] 2) 资源簇内查找:首先依据QA的类型及属性值信息查询区间划分表,确定查询资源的笛卡尔坐标,并转化为HSFC编码;然后依据Chord规则定位到提供资源的节点;最后将资源及节点的IP发送给用户,用户可以直接与该节点通信。

[0032] 3) 如果资源节点提供的资源不足或者无此资源,资源节点将区间邻居节点的资源反馈给用户并附上邻居节点标记(表示资源来源于邻居)。如果区间邻居节点也没有资源,那么可以再提供邻居的区间邻居节点的资源(这个间接度可以自由设定)。如果还是没有资源则向用户报告查找失败。

[0033] 4) 用户集合资源节点提供的资源信息。当资源来自多个资源簇时,按匹配权重排序,提供给用户。

[0034] 虽然保持区间邻居节点信息会增加节点的存储开销,但该系统建立在云服务器上,存储容量已不再是主要限制因素。却能带来简化查询、提高查找效率等便利。考虑到存在节点加入退出等情况,资源首次登记时,节点的区间邻居信息只存储HSFC码,而在第一次查找和固定维护期间,节点才会维护自己的邻居,存储每个HSFC码对应的节点IP;而当节点需要将数据转移时,会主动通知邻居节点更新邻居节点IP。

[0035] 本发明能够在计算复杂度为对数级的查询路由跳数定位云资源数据。在分层云对等网络的基础上,结合资源簇定位方法、资源簇内的HSFC编码方案及匹配权重计算方法快速而准确地获取云系统中的目标资源,并利用资源簇内资源空间分布的邻居节点进一步优化资源查询。使反映查询路由跳数的时间复杂度、索引层节点的索引信息的空间复杂度及资源簇内节点邻居信息存储的空间复杂度的分别为 $O(\log m + \log \frac{n}{m^2})$, $O(\log m)$, $O(k)$, 其中 n 为网络节点数, m 为资源簇数, k 为资源簇内的类型数。

[0036] HChord算法默认所有的网络资源包含资源类型和资源属性值两种属性。并采用双

层Chord模型,节点的加入与离开遵循Chord的规则。该算法适用于一般的P2P网络和云网络。对于一般的P2P网络,节点变动较大,在线稳定、性能优秀的节点作为资源簇的代理节点(即其所代表的资源簇入口)。如图1所示实施例HChord网络的基本模型图。第一层实现资源类型信息的索引,将相关类型的资源聚集在一起,形成资源簇,也就是第二层。通过对资源类型的匹配,直接锁定可以存储和提供资源的资源簇,然后在规模较小资源簇内依据资源属性值查找特定资源。

[0037] 1、资源簇定位

[0038] 资源簇负责存储最终资源或资源信息,而其存储的资源类型组合便标识了该资源簇。资源簇是动态生成的,其包含的属性个数以及查询语句的关键字都是随机的,因此,能够有效定位资源簇是一个关键。当查询关键字不等于资源簇的类型组合时,直接哈希查询关键字并不能定位资源簇,而使用洪泛法能定位资源簇,但是效率低、网络通信量大。由于目前存储器容量已不再是主要限制因素,因此可以适当增加资源簇的索引而提高资源查找速度。

[0039] 索引层上的节点代表着资源簇的入口,同时也存储着一些指向其他资源簇代理节点的索引项。索引项的简单数据结构包括索引向量和相应资源簇代理节点的IP地址,如图2中资源簇索引示意图,(a, b, c)表示索引向量,IP(Node_{abc})表示资源簇代理节点的IP地址。索引向量代表资源簇所提供资源的类型组合,是由类型关键字组成,并且关键字按字典序排序。例如某资源簇提供关键字为a, b, c三种类型的资源,则该资源簇的索引向量为(a, b, c)。

[0040] 正如上文所说,所有的网络资源都需要包含资源类型和资源属性值两种属性。在定位资源簇时,就需要使用资源类型作为索引。索引建立方法如下:

[0041] A:假如有某一资源S的索引向量为(a, b, c),利用SHA-1散列函数分别求哈希值Hash(a),哈希值Hash(b),哈希值Hash(c),哈希值Hash(abc);

[0042] B:利用Chord规则根据A中的哈希值确定相应节点Node_a(Hash(a)的责任节点,后面的节点同理)、Node_b、Node_c和Node_{abc};

[0043] C:上面这些节点都包含了存储S的资源簇的索引项,如图2所示。而Node_{abc}中的索引(a, b, c)指向了节点本身(用Local表示),因为该节点本身就是资源簇的入口节点。

[0044] 如图2所示,当节点Node_a查询包含的索引向量为(a, b)时,由于没有索引向量为(a, b)的资源簇,所以不能直接哈希定位资源簇;但(a, b)是包含在(a, b, c)中,通过哈希类型a或b时,能够间接得到Node_{abc}的地址信息,而查询的代价只是索引层节点的索引信息空间复杂度 $O(\log m)$,其中 m 为资源簇数。这就是HChord系统的资源簇定位算法。

[0045] 为了能让该算法更有效地工作,以下从索引的扩展性、维护等方面进行分析和改进:

[0046] (1) 每当有 k 维度资源(资源属性个数为 k ,其中 $k > 1$)登记时,在索引层就有 $k+1$ 份相同的索引存在。由于资源的维度增长很缓慢,大部分的资源簇类型维度都较低,因此索引层有多份相同的索引并不会增加系统的存储负担,反而给系统提供更好的容错性。正因为资源维度都较低,所以容易形成了下表1所示的情形:节点Node_a的索引项较多,但索引中包含的资源类型都较少,这既占用存储空间又表示系统中有太多的小型资源簇。而如果将资源簇ab,ac都融合到资源簇abc中如表2所示,则能减少索引所占空间和资源簇数量,也能

提高簇的热度和降低簇间查找概率,提高查找效率。当新索引项记录在索引表时,新索引项的关键字与索引表里的其他索引项比较,如果包含它,就融合它的资源簇。

[0047] 表1

a	
ab	IP(Node _{ab})
ac	IP(Node _{ac})
abc	IP(Node _{abc})
...	...

[0048]

[0049] 表2

a	
abc	IP(Node _{abc})
...	...

[0050]

[0051] (2) 当资源簇的代理节点发生变化时,会引发相应节点更新索引。由于可以并发更新索引,所以更新的时间复杂度约为 $O(\log m)$ 。而且由于该系统可部署在云对等网络,索引层节点稳定性好,索引更新对整个系统的影响可以忽略不计。在HChord系统运行的初期,由于可能存在许多类型维度较低的资源簇,在资源簇的融合过程中会引起HChord系统的波动,但也不会太大地增加查询延迟。资源簇的类型维度增长到一定程度就会趋于稳定,这时整个系统的资源簇融合也就很少发生。

[0052] 2、HSFC编码

[0053] 通过建立资源类型索引可以方便定位资源簇,然后再建立资源属性值索引,便可定位资源节点。由于资源属性值可能是整数、小数或者是字符类的,如果以云资源属性值向量直接哈希存储,会增加检索的复杂性,同时也会破坏原有数据的一些相关性,降低检索效率。因此该文先将每个类型的属性值分段,在逻辑上形成一个多维资源空间;然后利用空间填充曲线将多维资源降维,使之映射到单维资源环上。

[0054] 首先,将每个特定的属性值向量转化为相应的资源空间笛卡尔坐标。每个资源簇就是一个资源空间,资源簇内每一类型就是资源空间的一个维度,k维类型就形成一个k维笛卡尔坐标系。每个类型的属性值以其值域被划分为d个区间,这样便形成了 d^k 个多维区间。每个多维区间的属性值向量都可以用这个多维区间的笛卡尔坐标值 $D(x_1, x_2, \dots, x_k)$ 来表示。由于类型个数和区间划分情况都可能是变动的,因此每个资源簇保存一张区间划分表(Range table)。依据区间划分表,属性值向量能转化为相应的笛卡尔坐标。如图3所示实施例HChord网络资源簇内的资源分布空间及Hilbert环示意图:资源簇包含2个类型:A和B,每个类型划为4个区间,其中属于阴影部分的属性值向量都可以 $D(11, 01)$ 表示(或写为 $(1101)D$),其中坐标值用二进制表示。

[0055] 然后,将笛卡尔坐标值转化为相应的HSFC码,组成一个Hilbert环,以实现多维键

值向单维键值的转化。如图3所示, (0000) D=(0000) HSFC, (1100) D=(1111) HSFC。平面内的粗曲线即是將二维笛卡尔坐标转化为HSFC值所形成的一维Hilbert曲线, 实现了降维的目的。然后再將Hilbert曲线首尾相接, 覆盖在Chord环上, 组成一个Hilbert-Chord环。节点的加入和离开还是依照Chord的规则, 而资源的检索则结合了HSFC码和Chord的相关规则。

[0056] 3、匹配权重计算方案

[0057] 当有多个资源簇满足查询时, 来自不同资源簇的查询结果的匹配权重可能不同。那些满足查询且类型维度更低的资源簇具有更高的匹配权重。类似的, 资源簇内的资源节点提供的资源也比其区间邻居节点所提供的更准确, 匹配权重也就更高。具体的公式如下:

[0058] 假设某资源簇的类型维度为C, 查询语句的类型维度为 C_q , 则类型的匹配权重为:

$$[0059] \quad W_1 = \frac{C_q}{C}$$

[0060] 假设资源簇内的某资源节点的邻居跳数为X(当资源来自节点本身, X为0; 来自直接邻居, X为1, 以此类推), 则属性值的匹配权重为:

$$[0061] \quad W_2 = \frac{1}{1+X}$$

[0062] 最终的匹配权重为:

$$[0063] \quad W = \alpha W_1 + \beta W_2$$

[0064] 其中, α 为类型匹配因子, β 为属性值匹配因子, 可由人工配置。

[0065] 当W越大, 匹配程度越高。而当资源簇的类型维度与查询的类型维度相差越大, 或者查询结果来源于原资源节点的邻居节点, 则W越小, 匹配程度也就越低。

[0066] 4、区间邻居

[0067] 一个k维的资源簇空间会被划分 d^k 个多维区间。除了在空间边界上的多维区间, 其他的多维区间在任意维度上都有两个相邻的区间, 这些区间便是邻居区间。存储邻居区间资源的节点被称为区间邻居节点。每个资源节点都会存储其区间邻居节点的索引信息, 空间复杂度约为 $O(k)$, k为类型维度。根据局部性原理, 区间邻居节点所存储的资源往往与原节点的是相关的。对于查询未命中或者某些复杂查找, 这些索引信息能为查询操作提供更多的选择。如图3所示, 阴影多维区间(1101) D的邻居区间有(1100) D、(1001) D和(1110) D。由于在空间边界上, 故在A1维度上少了一个邻居区间。

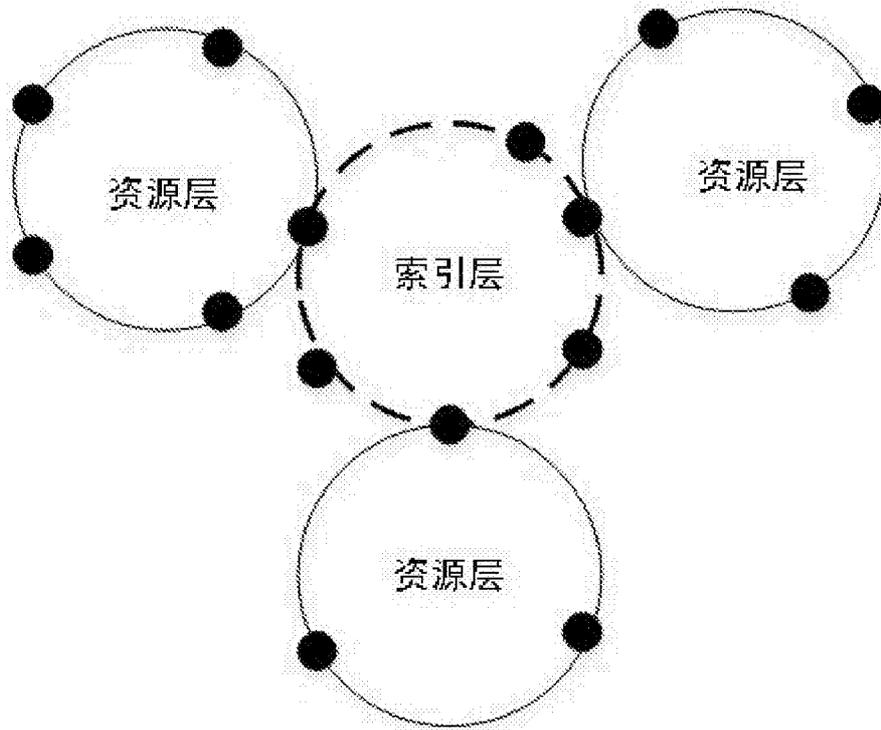


图1

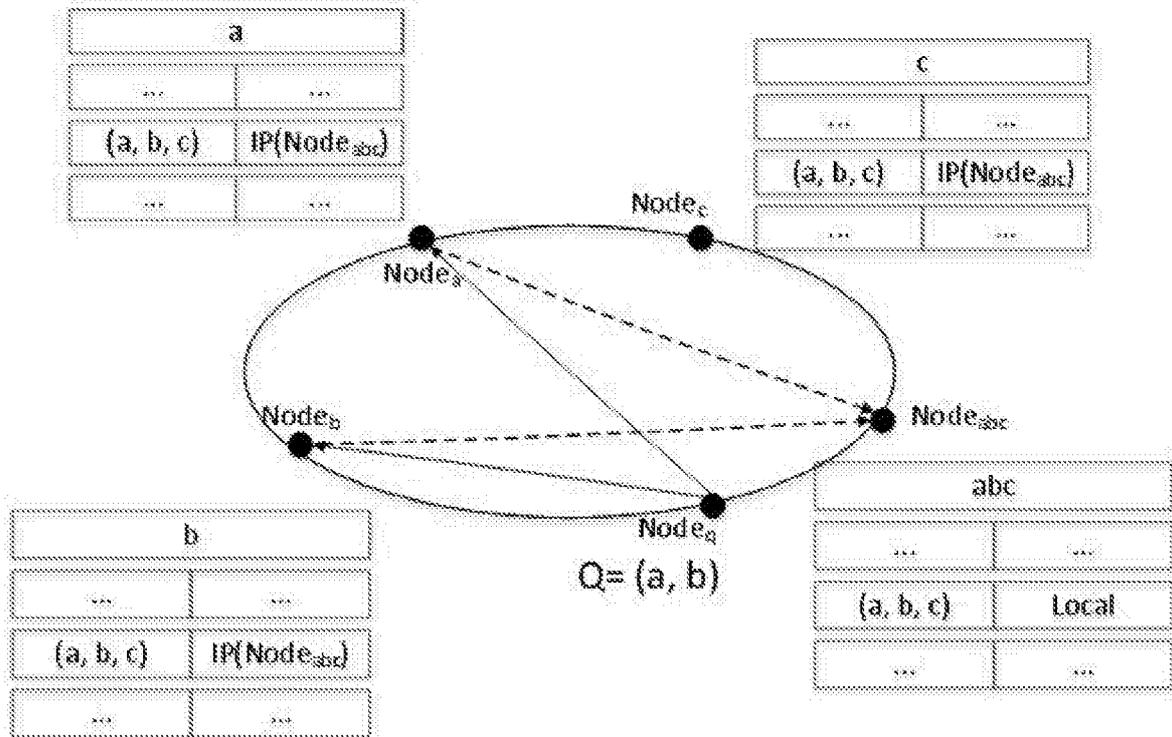


图2

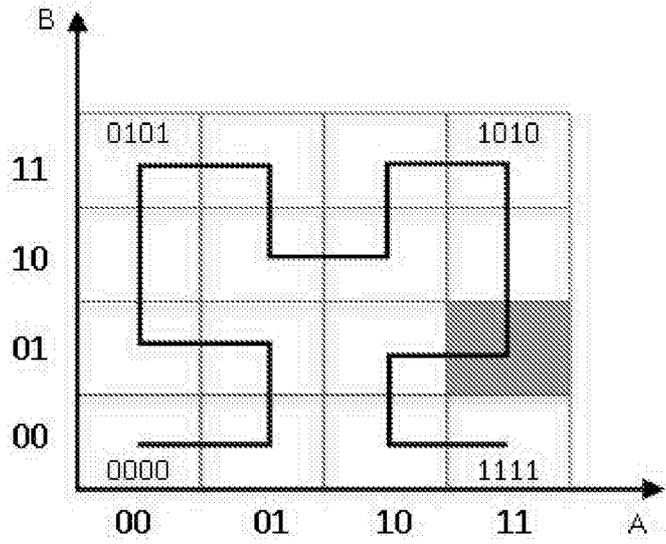


图3