一种面向结构化 P2P 网络的基于闲谈的资源发现方法

邓 泽[1],冯 丹[2],周 可[2],施 展[2]
(1. 中国地质大学(武汉)计算机学院,湖北武汉 430074;2. 华中科技大学计算机科学与技术学院,湖北武汉 430074)

摘 要: 结构化 P2P 网络下的多属性资源发现一直是一个公开问题.本文针对当前一种新颖的、优于传统方法的多属性资源发现方法 – PIRD,深入分析了其在网络动态变化时可能出现的低查询效率问题,并提出一种解决方法:基于闲谈的 PIRD(Gossip-based PIRD,G-PIRD).G-PIRD 通过闲谈算法估计网络规模,动态调整资源索引的发布以保证高的查询效率.同时针对 G-PIRD 可能导致的负载不均衡问题,提出一种基于有界 LSH(Bounded LSH,B-LSH)的负载均衡策略.试验证明:G-PIRD 能动态适应网络变化,保证高效率的多属性资源发现;以及 G-PIRD 的负载均衡策略在保证高查询效率的同时,大大地降低了节点的索引负载.

关键词: 结构化 P2P 网络; 多属性资源发现; 闲谈算法; 负载均衡
中图分类号: TP393 **文献标识码:** A **文章编号:** 0372-2112 (2010) 11-2510-08

A Gossip-Based Approach for Resource Discovery in Structured Peer-to-Peer Networks

DENG Ze[1], FEND Dan[2], ZHOU Ke[2], SHI Zhan[2]
(1. *School of Computer, China University of Geosciences, Wuhan, Hubei* 430074, *China*;
(2. *School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, Hubei* 430074, *China*)

Abstract: Multi-attribute resource discovery in structured Peer-to-Peer (P2P) networks is still an open problem. Recently, a novel and more efficient multi-attribute resource discovery approach than traditional ways — P2P-based Intelligent Resource Discovery (PIRD) has been proposed. However, PIRD may be inefficient under network churns. To address this issue, in this paper, a gossip-based PIRD (G-PIRD) is proposed. G-PIRD employs a gossip algorithm to learn the estimated value of network size and dynamically publishes resource indexes to keep a great efficiency of resource discovery. Meanwhile, a load balancing scheme based on bounded LSH (B-LSH) is proposed to deal with the problem of potential load imbalancing of G-PIRD. Extensive experiments show that G-PIRD can adapt well to the changes of network size to maintain high query efficiency and the proposed load balancing scheme can greatly reduce the maximum number of indexes per node with the slight loss of query efficiency.

Key words: structured P2P networks; multi-attribute resource discovery; gossip algorithm; load balancing

1 引言

由于结构化 Peer-to-Peer(P2P)网络(如 Chord[1]、CAN[2]和 Pastry[3])的高查询效率和高可扩展性,即对于一个包含 N 个 peers 的结构化 P2P 网络,查询效率是 $O(\log N)$跳而每个节点只需维护 $O(\log N)$个邻居的信息[4],大量的资源发现系统和信息检索系统建立在结构化 P2P 网络之上.如:Twine[5]、pSearch[6]、SIPPER[7]和 SPRITE[8]等等.在实际应用中,这些系统被要求支持对资源的多属性匹配查询.例如,在 Twine 这个资源发现系统中,用户会要求发现"Memory = 1GB ∧ CPU = 2GHz"的计算资源用于计算.然而,结构化 P2P 网络面临的重大挑战之一是它不能有效率地支持各种复杂查询,其中包括多属性查询[9].

为了有效率地在结构化 P2P 网络中进行多属性资源发现,本文针对当前一种优于传统方法的多属性资源发现方法 – P2P-based Intelligent Resource Discovery, PIRD[16],深入分析了其在网络动态变化时可能出现的低查询效率问题,并基于一种在分布式环境下收集全局信息的闲谈算法[17],提出基于闲谈的 PIRD (Gossip-based PIRD,G-PIRD)以改善 PIRD 的查询效率.同时针对 G-PIRD 可能导致的负载不均衡问题,提出一种基于有界 LSH(Bounded LSH,B-LSH[18])的负载均衡策略.

收稿日期:2009-08-25;修回日期:2009-11-03
基金项目:国家 973 重点基础研究发展规划(No.2004CB318201);国家 863 高技术研究发展计划(No.2009AA01A402)

2　相关工作

对于结构化 P2P 网络下的多属性资源发现问题,传统的解决方法有两类.一类如文献[10~12].这类方法把一个包含 k 个查询关键字的查询分拆为 k 个子查询语句,分别进行查询,最后合并这些查询结果.由于一条查询被拆分成多条查询,产生大量的中间查询结果,这会导致了高的网络查询开销.另一类方法如文[13,14].它们对一个包含 m 个属性的资源,进行多个属性的组合,根据这些属性组合的键值对资源进行索引,并发布这些索引到网络中以支持多属性查询.这类方法的问题是潜在的高索引数目导致高索引维护开销.也就是说,最坏情况下,一个资源的索引数目是 $\binom{m}{1}+\binom{m}{2}+\cdots+\binom{m}{m}=2^m-1$,趋向指数增长.大量的索引会导致高的维护开销.

最近,Haiying Shen 等提出一种基于局部性敏感哈希(Locality Sensitive Hashing,LSH[15])的多属性资源发现方法(P2P-based Intelligent Resource Discovery,PIRD[16]).PIRD 把一个包含 m 个属性的资源看成一个资源对象,利用 LSH 对这个资源对象进行 L 次索引,由于 LSH 的局部性敏感特征,相同或相似的资源被索引到结构化 P2P 网络中空间位置相同或相近的节点上.这样,当节点接收到一个包含多个查询关键字的查询时,通过 LSH,查询同样被转发到存放与其有相同或相似关键字组合的资源索引的节点上.相对于上述的第一类传统方法,PIRD 不需要对查询语句进行拆分,保证了低的网络查询开销.相当于第二类传统方法,当一个资源包含的属性数目 m 很大时,PIRD 发布的索引数目 $L\ll 2^m-1$,故 PIRD 能在保持高查询效率的同时,大大地减少索引开销.但由于 PIRD 没有充分考虑分布式环境下的全局信息(总节点数)的变化,导致当网络动态变化时,PIRD 可能出现低资源发现效率的问题(详见 4.1 节).

3　基于对等网络的智能资源发现(P2P-based Intelligent Resource Discovery,PIRD)

为了更好地描述 PIRD 和提出 G-PIRD,首先给出与多属性资源发现相关的定义:

定义 1　(结构化 P2P 网络):本文的结构化 P2P 网络被定义为一个被广泛接受和应用的 Chord 环[1].环中所有节点根据其节点标示符(Identifier,ID)值,由小到大顺时针连接形成一个环.

定义 2　(资源):一个包含 s 项属性的资源 r 被定义为一个属性集合 $RS=\{ra_i\},\forall i\in[1,s]$.其中,$ra_i$ 表示资源的第 i 项属性.

定义 3　(资源向量):资源 r 的属性集 $RS=\{ra_i\}\ \forall i\in[1,s]$被一个长为 d 比特的位向量表达.向量的产生过程:定义一个 d 比特长的向量 v 并初始化所有位为"0";通过一个均匀分布特征的哈希函数,映射每个属性 ra_i 到 v 中的某一位,并置"1".形成的位向量 $r.v$ 被定义为 r 的资源向量.

定义 4　(资源索引):资源 r 的一条索引被定义为 $in=<key,(r,location(r))>$.其中,key 表示资源的特征信息(一个属性或一组属性)在节点 ID 空间中的映射值,被用于定位 Chord 环中存放该 in 的节点;r 同定义 2;$location(r)$表示拥有资源 r 的节点的 IP 地址信息.

定义 5　(查询):一个包含 t 项属性匹配要求的查询 q 被定义为一组被要求匹配的属性 $QS=\{qa_j\},\forall j\in[1,t]$.其中,$qa_j$ 是查询的第 j 项被要求匹配的属性.

定义 6　(查询向量):查询 q 的属性要求集 $QS=\{qa_i\}\ \forall i\in[1,t]$被一个长为 d 比特的向量表示.向量的产生过程:定义一个 d 比特长的向量 v 并初始化所有位为"0";通过一个均匀分布特征的哈希函数,映射每个被查询属性 qa_i 到 v 中的某一位,并置"1".形成的位向量 $q.v$ 被定义为 q 的查询向量.

定义 7　(资源完全匹配查询):给定一个查询 q 和一个资源 r,当 $q.QS\subseteq r.RS$ 时,r 完全匹配 q,并被定义为 $r\doteq q$.

定义 8　(资源相似匹配查询):给定一个查询 q 和一个资源 r,设定一个距离函数 D 和距离阈值 d,当 $q.QS\not\subset r.RS$ 且 $q.QS\cap r.RS\neq\emptyset$ 且 $D(q.v,r.v)\leqslant d$ 时,r 相似匹配 q,并被定义为 $r\cong q$.

PIRD 被用于在结构化 P2P 网络中,发现与查询完全或相似匹配的资源(见定义 7 和 8).其基本原理:把有相同属性和相似属性资源的索引(定义 4)发布到结构化 P2P 网络中的标识符空间位置上相同和相近的节点上.这样当有查询发生时,采用与发布索引时相同的方法,把查询转发到对应的节点上,并通过这些节点所存放的资源索引,获取与其查询完全或相似匹配的资源.如图 1,PIRD 的基本架构包括三个部分:局部性节

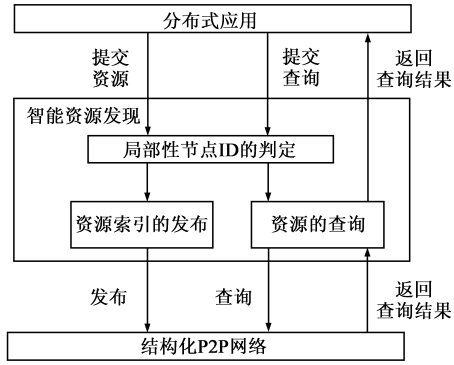

图1　PIRD的基本架构

点 ID 的判定;资源索引的发布;和资源的查询.

3.1 局部性节点 ID 的判定

PIRD 利用一种基于局部性敏感的哈希(Locality Sensitive Hashing,LSH)函数[15]来判定资源或查询所对应的节点 ID.一个 LSH 函数族被定义如下:

定义 9 (LSH 函数族):定义一个哈希函数族 $H=\{h:S\to U\}$.设 S 表示点的集合域,U 表示整数域,D 表示计算两点之间距离的距离函数.对于给定的两个距离阈值 $d_1,d_2(d_1<d_2)$和两个概率值 $p_1,p_2(p_1>p_2)$如果任意两点 $x_1,x_2\in S$,满足下列两条件:

—如果 $D(x_1,x_2)\leqslant d_1$,则 $Pr_H[h(x_1)=h(x_2)]\geqslant p_1$.

—如果 $D(x_1,x_2)\geqslant d_2$,则 $Pr_H[h(x_1)=h(x_2)]\leqslant p_2$.

则 H 是一个基于距离函数 D 的对(d_1,d_2,p_1,p_2)敏感的哈希函数族.

不同的距离函数 D 产生不同的 LSH 函数族.PIRD 选择基于 p-stable 分布(如 Gaussian 分布)[19]的 l_2 范数(即 Euclidean 范数)[20]为距离函数 D,故 LSH 函数族中的每个函数被表示为:

$$h_{a,b}(\boldsymbol{v})=\left\lfloor\frac{\boldsymbol{a}\cdot\boldsymbol{v}+\boldsymbol{b}}{w}\right\rfloor \tag{1}$$

其中 $\boldsymbol{v}$ 是一个 d 维向量,$\boldsymbol{a}$ 是一个基于 p-stable 分布的与 $\boldsymbol{v}$ 同维的随机向量,$\boldsymbol{a}\cdot\boldsymbol{v}$ 表示两个向量的点乘.w 是一个给定的正实数,b 是$[0,w]$的一个随机实数.通过公式(1),d 维的资源向量 $\boldsymbol{v}^d$ 被映射到一个整数域 Z 中 $\boldsymbol{v}^d\to Z$.

基于上述所定义的 LSH 函数簇,PIRD 判定一个资源 r 或一个查询 q 所对应节点 ID 的过程如下:

(1)根据接收到资源或查询,计算资源向量 $\boldsymbol{r}.\boldsymbol{v}$(见定义 3)或查询向量 $\boldsymbol{q}.\boldsymbol{v}$(见定义 6).

(2)定义一个函数族 $G=\{g:\boldsymbol{v}^d\to U^M\}$,用于把一个 d 维向量映射为 M 个整数.G 中的每个函数被表示为:

$$g(\boldsymbol{v})=(h_1(\boldsymbol{v}),\cdots,h_M(\boldsymbol{v})),(h_i(H,1\leqslant i\leqslant M) \tag{2}$$

从 G 中随机选择 L 个函数 $g_j(\boldsymbol{v})$ $(1\leqslant j\leqslant L)$,产生 L 个整数集合 $A_j=\{a_1^j,\cdots,a_M^j\}(1\leqslant j\leqslant L)$.$L$ 值的判定依照如下公式:

$$L=\left\lceil\frac{\log 1/\delta}{-\log(1-p_1^M)}\right\rceil \tag{3}$$

其中 δ 表示至少能以 $1-\delta$ 的概率值的概率发现查询范围 d_1 内的所有资源.p_1见定义 9,并通过公式(4)可得:

$$p_1=\int_0^w f_s(t)(1-t)\mathrm{d}t \tag{4}$$

其中,$f_s(t)$是 p-stable 分布的概率密度函数,w 同公式(1)中的 w.

(3)定义一个 ID 判定函数 f:

$$f(a_1,\cdots,a_M)=((\sum_{i=1}^{M} random_i\times a_i)\ \mathrm{mod}\ prime)\ \mathrm{mod}\ ID_{space} \tag{5}$$

其中,$a_1,\cdots,a_M$ 是一组整数,$random_i$ 是一个随机数,$prime$ 是一个足够大的素数,ID_{space}是 P2P 网络节点的标识符空间的大小.根据步骤(2)中得到的 L 个整数集合 $A_j=\{a_1^j,\cdots,a_M^j\}$ $(1\leqslant j\leqslant L)$,通过 f 产生 L 个 IDs.

3.2 资源索引的发布和资源的查询

当 PIRD 接收到一个资源 r 时,通过 ID 判定方法得到 L 个 IDs,并产生 L 个资源索引 $in_i=<ID_i,(r,location(r))>(1\leqslant i\leqslant L)$(见定义 4).PIRD 通过结构化 P2P 网络的协议规范(这里特指 Chord),分别发布 in_i 到 Chord 环中其节点 ID 值大于且最接近 ID_i 的节点(也就是 ID_i 的后继节点,详见 Chord[1])上存放.

当 PIRD 接收到一个查询 q 时,通过同样的 ID 判定方法得到 L 个 IDs,并转发 q 到 ID_i 的后继节点上查找匹配的资源.同时由于相似资源的索引被发布到相邻近的节点上,q 也被转发到距离 ID_i 的后继节点一定跳数内的邻近节点上查找匹配的资源.当某个被查询节点接收到 q 时,初始化查询结果集 $result=\emptyset$.假设该节点上存放了 t 个资源索引,则计算得对应的资源向量 $r_1.v,\cdots,r_t.v$.通过 l_2 范数距离函数 D,计算查询向量 $\boldsymbol{q}.\boldsymbol{v}$ 和这些资源向量之间的距离:

$$D(\boldsymbol{q}.\boldsymbol{v},\boldsymbol{r}.\boldsymbol{v})=\|\boldsymbol{q}.\boldsymbol{v}-\boldsymbol{r}.\boldsymbol{v}\|=\sqrt{\sum_{i=1}^{d}(\boldsymbol{q}.\boldsymbol{v}_i-\boldsymbol{r}.\boldsymbol{v}_i)^2} \tag{6}$$

当 $D(\boldsymbol{q}.\boldsymbol{v},\boldsymbol{r}_i.\boldsymbol{v})\leqslant$阈值 d_1 时$(1\leqslant i\leqslant t)$,说明 $r_i\doteq q$ 或 $r_i\cong q$,则更新 $result=result\cup\{(r_i,location(r_i))\}$.匹配完毕后返回 $result$ 至查询发起节点.

4 基于闲谈的 PIRD(Gossip-based PIRD,G-PIRD)

4.1 PIRD 的问题分析

基于上节内容可知,PIRD 的核心是通过局部性节点 ID 的判定,把有相同属性和相似属性资源的索引发布到结构化 P2P 网络中的相同和距离相近的节点上.但是,实际上 PIRD 不一定能保证相似资源的索引被发布到跳数距离相近的节点上.

这个问题通过图 2 进行说明.假设 PIRD 发布两个相似资源 r_1,r_2 的索引到一个 ID 空间 $\mathscr{R}=[0,2k-1]$ 的 Chord 环中$(k=128)$.进一步假设通过局部性节点 ID 的判定过程后,r_1 对应的一个 ID = 0,r_2 对应的一个 ID = 11.图中,黑实心圈表示 r_1,r_2 分别对应的 ID,大圈表示在空间[0,12]中实际存在的节点.

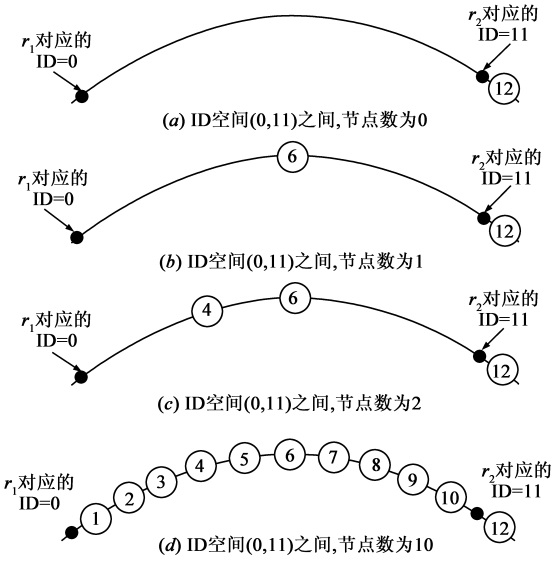

图2　PIRD的问题图示

图 2(*a*)反映当环中节点数目很少时,ID 空间[0, 12]之间只有一个 ID = 12 的节点存在,这时根据索引的发布原则(即放置到资源对应 ID 的后继节点上),r_1 和 r_2 的索引都被放置到节点 12 上.这是希望看到的结果.图 2(*b*)、2(*c*)显示,当空间[0,12]之间节点数目有所增加时,同样根据索引的发布原则,r_1 的索引分别放置到节点 6 和节点 4 上,而 r_2 的索引依然放置在节点 12 上.这时,放置 r_1 索引的节点和放置 r_2 索引的节点之间的距离分别是一跳、和两跳.这是也是希望看到的结果,因为放置 r_1 索引的节点,可以通过其维护的称之为 *finger table* 的路由表中的 *successor* 项(详见 Chord[1]),以一跳、和两跳的代价路由查询到放置 r_2 索引的节点上.而放置 r_2 索引的节点,也可以通过其 *finger table* 中的 *predecessor* 项(详见 Chord[1])以一跳、和两跳的代价路由查询到放置 r_1 索引的节点上.但是,当网络规模进一步增大时,如图 2(*d*)所示,放置 r_1 索引的节点 1 和放置 r_2 索引的节点 12 之间的距离达十跳.这会导致大的网络开销以获得 r_1 和 r_2 的索引.而从查询效率方面考虑,如果给定的最大查询跳数距离小于十跳时,查询被转发到节点 1 获得 r_1 的索引后,不可能再到达节点 12 以获得 r_2 的索引.

导致上述问题的原因是:PIRD 定义的 ID 判定函数 f(见公式(5))只是从 ID 空间距离角度考虑两个相似资源产生相近的 ID 值.具体而言,设环上任意两点 x, y 之间的距离被定义为:

$$Dist(x,y)=\begin{cases} y-x, & (x<y) \\ 2^k-(x-y), & (x\geqslant y) \end{cases} \tag{7}$$

则通过 f 产生的两个资源 r_1 和 r_2 对应的标识符 ID_1、ID_2 之间的空间距离为 $Dist(ID_1, ID_2)$.

由于 ID 空间 $\mathscr{R}$ 的大小 $ID_{space}=2^{128}-1$,是固定的.且 Chord 中节点 ID,是通过一个均匀分布特征的哈希函数产生的,故当网络规模为 N 时,则 ID 空间的单位节点密度为:

$$Density=\frac{N}{ID_{space}} \tag{8}$$

故,ID_1, ID_2 之间的平均跳数为:

$$\begin{aligned} Hop(ID_1,ID_2) &= Dist(ID_1,ID_2)\times Density \\ &= Dist(ID_1,ID_2)\times\frac{N}{ID_{space}} \end{aligned} \tag{9}$$

可以看到,在给定 $Dist(ID_1, ID_2)$下,ID_1, ID_2 之间的跳数距离 $Hop(ID_1, ID_2)$会随着网络规模 N 的增减而增减.设放置相似资源索引的节点之间的最大跳数距离为 MAX_{hop},当 $Hop(ID_1, ID_2) > MAX_{hop}$时,PIRD 会因为不能查找到所有放置匹配资源的节点而导致查询效率下降.

4.2　解决方法:G-PIRD

基于上述问题分析可知,问题解决的关键是使 ID 判定函数 f 从跳数距离角度考虑,产生资源的对应 ID.下面讨论如何改进 ID 判定函数 f.不妨假设 f 已经被改进,也就是说判定函数 f 能从低跳数距离角度考虑,产生资源对应的 ID.则通过 f 产生的两个资源 r_1 和 r_2 对应的标识符 ID_1、ID_2 应满足如下不等式:

$$Dist(ID_1,ID_2)\times Density\leqslant MAX_{hop} \tag{10}$$

也就是

$$Dist(ID_1,ID_2)\leqslant MAX_{hop}\times\frac{ID_{space}}{N} \tag{11}$$

设 $ID_2 > ID_1$,则基于公式(5),不等式(11)左边的 $Dist(ID_1, ID_2)$可表示为:

$$\begin{aligned} Dist(ID_1,ID_2) &= ID_2-ID_1=f(d_1,\cdots,d_M)-f(c_1,\cdots,c_M) \\ &= \Big(\Big(\sum_{i=1}^{M} random_i\times(d_i-c_i)\Big)\bmod prime\Big) \\ &\quad \bmod ID_{space} \end{aligned} \tag{12}$$

$c_1,\cdots,c_M$ 和 $d_1,\cdots,d_M$ 是根据公式(2)产生的分别对应 r_1 和 r_2 的整数集合.设 $random'$ 是随机数集 $random_1$, $\cdots$, $random_M$ 的均值,则根据公式(1)、(12)可得:

$$\begin{aligned} Dist(ID_1,ID_2) &\approx \Big(\Big(random'\times\sum_{i=1}^{M}\Big(\frac{a_i\cdot(r_2\cdot v)+b_i}{w}-\frac{a_i\cdot(r_1.v)+b_i}{w}\Big)\Big)\bmod prime\Big)\bmod ID_{space} \\ &= \Big(\Big(\frac{random'}{w}\times\sum_{i=1}^{M}a_i\cdot(r_2.v-r_1.v)\Big) \\ &\quad \bmod prime\Big)\bmod ID_{space} \end{aligned} \tag{13}$$

设 r_1 和 r_2 之间不同属性的数目为 c,则根据资源向量的定义(见定义 3)可知资源向量 $\boldsymbol{r_1.v}$ 和 $\boldsymbol{r_2.v}$ 之间的不同项的数目为 $2c$.另一方面,由于公式(13)中的 a_i(1≤

$i \leqslant M$)是基于 p-stable 分布的随机向量,不妨设该向量中的每项为 p-stable 分布的期望值 e.则可得:

$$\sum_{i=1}^{M} a_i \cdot (r_2.v - r_1.v) \approx 2c \times e \times M \tag{14}$$

将公式(14)带入公式(13)中,可得

$$Dist(ID_1, ID_2) \approx ((\frac{random'}{w} \times 2c \times e \times M) \bmod prime) \bmod ID_{space} \tag{15}$$

进一步基于公式(11)、(15),可得:

$$((\frac{random'}{w} \times 2c \times e \times M) \bmod prime) \bmod ID_{space} \leqslant MAX_{hop} \times \frac{ID_{space}}{N}$$

故

$$random' \leqslant ((MAX_{hop} \times \frac{ID_{space}}{N} \times \frac{w}{2c \times e \times M}) \bmod prime) \bmod ID_{space} \tag{16}$$

通过上述分析可以看到,当 ID 判定函数 f(公式(5))中的 $random_i$ 随机取值满足不等式(16)时,可保证有 c 个不同属性的资源能至少以 $1-\delta$(见公式(3))概率值的概率,被索引到相近节点上.且这些相近节点之间的最远跳数距离为 MAX_{hop}.

但是,对于 P2P 分布式环境下的每个节点,不等式(16)右边公式参数中的网络规模 N 是不可知的.可以看到,如何让 P2P 分布式环境下中的每个节点知道 N 是改进 PIRD 的关键.在文[17]中,一种分布式环境下收集全局信息的闲谈算法(Gossip Algorithm)被给出.该算法可以保证在 $O(\log N)$次信息交互后,分布式环境下的每个节点可以获得全局信息(如总和、平均值等)的估计值.因此,这里应用该算法到 Chord 环境下以获得网络规模 N 的估计值.具体而言,Chord 中每个节点 i 维护一个二元组 $T_i = <sum_i, weight_i>$.其中 sum_i 表示本地维护的总和数据,$weight_i$ 表示该数据的权重.所设计的算法如下:

算法 1 Chord 环中收集节点规模信息的闲谈算法

(1)任意选择 Chord 环中的一个节点 s 初始化其二元组 $T_s = <1,1>$,其它所有节点 i 的二元组 $T_i = <1,0>(i \neq s)$.

(2)闲谈过程:对于每个节点 i 并行执行下面两步骤

步骤 1:节点 i 周期性执行如下过程 Γ 次,Γ 的上限为 $O(\log N)$

(a)从其 *finger table* 表中随机选一个 *finger* 节点 j 作为闲谈对象.

(b)发送二元组 $T_i = <sum_i, weight_i>$ 给 *finger* 节点 j.

(c)在接收到节点 j 的包含二元组 T_j 的响应消息后,更新本地 T_i.

$$T_i.sum_i = (T_i.sum_i + T_j.sum_j)/2,$$

$$T_i.weight_i = (T_i.weight_i + T_j.weight_j)/2$$

步骤 2:当节点 i 接收到一个来自节点 j 的闲谈消息后,执行如下过程

(a)发送包含本地二元组 T_i 的消息给节点 j.

(b)同步骤 1 中一样的方法,更新本地二元组 T_i

(3)每个节点通过本地 T_i 计算 N

$$N \approx \frac{T_i.sum_i}{T_i.weight_i} \tag{17}$$

通过算法 1 和公式(5)、(16),G-PIRD 中的每个节点可以根据网络规模的变化,动态地发布本地资源的索引到对应的节点上,以保证高查询的效率.

4.3 G-PIRD 的负载均衡策略

虽然 G-PIRD 解决了查询效率的问题,但公式(16)可能导致某些节点存放的索引数目过大而造成负载不均衡的问题,特别是当 MAX_{hop} 值设置过小时.为了解决这个问题,基于一种有界 LSH(Bounded LSH, B-LSH[18])的设计思想,设计 G-PIRD 的负载均衡策略.

B-LSH 能把具有非均匀分布特征的数据集均匀地映射到不同的哈希桶中,以减小存储空间开销和查询响应时间.具体而言,B-LSH 为每个桶设定一个最大容量 MAX_{cap}.当一个桶中放置的数据对象(由 d 维的向量表示)数目等于 MAX_{cap} 时,B-LSH 会计算出一个表示桶中数据对象平均值的中心数据对象 *center*.并通过距离函数 D(见公式(6))得到距离 *center* 最远的边界数据对象 *edge* 和距离 R_{edge}.如果有新的数据对象 u 被插入到这个桶中,当 $D(center, u) < R_{edge}$ 时,对象 *edge* 被对象 u 替换,并重新计算 *edge* 和 R_{edge}.对从桶中淘汰出的对象以相同的方式插入到相邻的桶中.如果 $D(center, u) \geqslant R_{edge}$,尝试插入 u 到相邻的桶中.如此循环,这样能保证每个哈希桶中有相近数目的数据对象.

在 G-PIRD 的应用环境下,哈希桶被节点代替,数据对象是资源向量.则 G-PIRD 的负载均衡策略如下:

算法 2 G-PIRD 的负载均衡算法

(1)每个节点 i 接收到一个资源索引 *index* 时,如果本地所存放的资源索引 $IN = \{in_k \mid 1 \leqslant k \leqslant MAX_{cap}\}$ 的数目 $|IN| < MAX_{cap}$,$IN = IN(\{index\})$,算法结束.否则转向(2).

(2)节点 i 计算本地的中心数据对象 *center*,边界数据对象 *edge* 和它们之间的向量距离 R_{edge}.

$$center = \frac{\sum_{k=1}^{MAX_{cap}} in_k.r.v}{MAX_{cap}} \tag{18}$$

$$edge = argmax\{D(in_1.r.v, center), \cdots, D(in_{MAXcap}.r.v, center)\} \tag{19}$$

$$R_{edge} = D(edge, center) \tag{20}$$

(3)如果 $D(index.r.v, center) \geqslant R_{edge}$,直接转向(4).否则 $IN = (IN - \{edge\ 对应的索引\}) \cup (\{index\}$.重新计算 *center*、*edge* 和 R_{edge}.把从集合 *IN* 中淘汰出的索引视作新接收到的资源索引 *index*,并转向(4).

(4)如果节点 *i* 的后继节点存储的索引数目 < MAX_{cap},转发 *index* 给其环中的后继节点.否则如果节点 *i* 的前驱节点存储的索引数目 < MAX_{cap},转发 *index* 给其前驱节点.否则随机转发 *index* 给这两个节点中的一个,且保证该接收节点不是 *index* 的发送节点.

5　试验结果

5.1　试验准备

为了评估 G-PIRD,一个基于 PeerSim[21] 和 eXist[22] 的资源发现仿真模型被开发.PeerSim 被用于产生一个任意网络规模的 Chord 环.eXist 是一个原生 XML 数据库,由于 eXist 是层次化目录管理 XML 文档,Chord 环中的每个节点被分配一个目录以存放本地资源信息和资源索引.资源信息和资源索引都由 XML 文档表示.

表 1　仿真参数表

参数	默认值
ID 空间大小 ID_{space}	$2^{128}-1$
节点数目 *N*	[500,4000]
资源数目 $NUM_{resource}$	10000
资源在网络中的分布	均匀分布
查询数目 NUM_{query}	100
查询半径 l_2范数距离 d_1	8
最远跳数距离 MAX_{hop}	8
不同属性数目 *c*	4
LSH 相关参数	
p-stable 分布的期望值 e	2
w	4
δ	0.1
M	4
L	5

试验数据集采用文[23]中的一个存储资源信息集合.该集合包含 10,000 个 XML 文档,每个 XML 文档描述一个包含 20 个属性的存储资源.为了体现资源的相似性,通过修改该数据集,保证每个资源有 12 个属性是相同的,其余 8 个属性的属性值随机产生.查询集合 S 包含 100 条多属性查询,所有查询都从 10,000 个资源信息中随机选取并抽取其属性而形成.具体的仿真环境和相关参数如表 1.

5.2　G-PIRD 与 PIRD 的查询性能比较

在这个试验中,比较 G-PIRD 和 PIRD 在不同网络规模和不同查询跳数距离情况下,包含 100 条查询的查询集合 *S* 所获的拟合召回率(*recall*,见公式(21)).

$$recall = \sum_{i=1}^{|S|} \frac{|B_i|}{\sum_{j=1}^{|S|} |A_j|} \tag{21}$$

其中 A_i 表示在查询半径 d_1 范围内,网络中完全或相似匹配查询 q_i 的所有资源;B_i 表示实际上从网络中发现的完全或相似匹配 q_i 的资源.

从图 3 中可以看到,当网络规模 $N = 500$ 时,PIRD 的查询性能在低查询跳数距离时略好于 G-PIRD.而随着查询跳数距离的增加,G-PIRD 和 PIRD 的查询性能是相当的.这是由于 PIRD 能保证相似资源的索引被放置到空间距离相近的节点上,当网络规模不大时,放置相似资源索引的节点之间的跳数距离很小,这样 PIRD 能从给定跳数距离范围内的节点中获取高的召回率.另一方面,G-PIRD 根据表 1 被设计为从查询跳数距离 $MAX_{hop} = 8$ 范围内的节点中,发现至少 90%(1 − δ)的完全或相似匹配的资源,而实际试验结果可以看到 G-PIRD 在查询跳数 $hop = 8$ 时,召回率 *recall* 为 93%,达到预期目标.

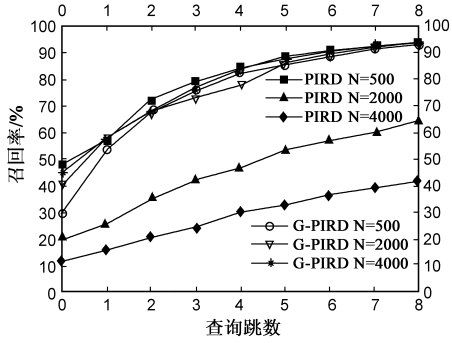

图3　PIRD和G-PIRD查询性能的比较

从图 3 中进一步可以看到,随着网络规模的增大,$N = 2000$ 和 $N = 4000$ 时,PIRD 的查询性能有很大的下降.如在查询跳数 $hop = 5$ 时,随着 *N* 从 500 到 2000、4000 时,对应的召回率 *recall* 从 87% 下降到 53%、33%.而 G-PIRD 在查询跳数 $hop = 5$ 时,随着 *N* 从 500 到 2000、4000 时,对应的召回率为 85%、86% 和 88%,查询性能没有明显变化.该试验结果证明了 G-PIRD 能够通过闲谈算法有效地改善 PIRD 对网络变化的自适应性,以保证高的查询效率.

5.3　G-PIRD 的负载均衡策略的评估

为了评估 G-PIRD 的负载均衡策略,首先测试在 $N = 2000$ 和不同资源数目情况下,G-PIRD 的负载均衡策略对节点存放的最大索引数目的影响.均衡策略中涉及到的参数 MAX_{cap} 通过公式(22)设定.

$$MAX_{cap} = \alpha \times NUM_{resource} \times L \tag{22}$$

其中 α 是一个比率,这里设定为 0.05;网络中资源数目 $NUM_{resource}$ 同样通过闲谈算法获取;*L* 见表 1.

图 4 显示 G-PIRD 的负载均衡策略大大地降低了节点存放的最大索引数目,特别是随着资源数目的增加.如在 $NUM_{resource} = 9000$ 和 10,000 时,存放的最大索引数目分别下降了约 67% 和 70%.

另一方面,测试负载均衡策略对 G-PIRD 的查询效率的影响.该试验在 $N = 2000$ 和 $NUM_{resource} = 10,000$ 情况下,对比 G-PIRD 实施负载均衡策略前后,查询集合 *S*

在查询半径 d_1(见表 1)范围内所获的拟合召回率.其测试结果如图 5,可以看到:G-PIRD 在实施负载均衡策略后,其查询效率略有下降.但依然能在 $MAX_{hop}=8$ 跳数范围内获得 92% 的高召回率,相对实施前,查询效率只下降了 2%.

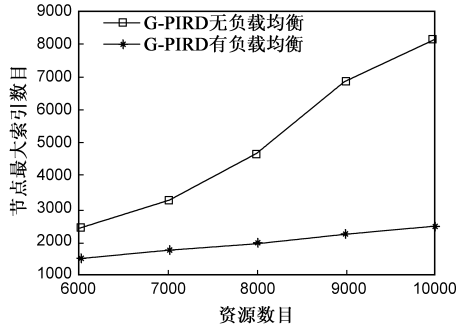

图4 在N= 2000和不同资源数目情况下, G-PIRD负载均衡策略的效果

通过上述两个测试结果可以看到,G-PIRD 的负载均衡策略在保证了高查询效率的同时,大大地降低了节点的索引负载,保证了 G-PIRD 的可扩展性.

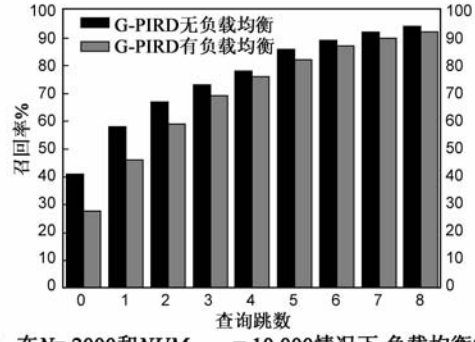

图5 在N= 2000和$NUM_{resource}$= 10,000情况下,负载均衡策略对G-PIRD查询效率的影响

6 结论

本文针对当前一种新颖的、面向结构化 P2P 网络的多属性资源发现方法 – PIRD,深入分析了其在网络动态变化时,可能出现的低查询效率的问题,并提出基于闲谈算法的 PIRD(G-PIRD)的解决方法.此外,进一步提出基于 B-LSH 的负载均衡策略以保证 G-PIRD 的可扩展性.试验证明:G-PIRD 能通过闲谈算法动态适应网络规模的变化,以保证高效率的多属性资源发现;G-PIRD 的负载均衡策略很大程度地降低了节点的索引负载,并保证了高的查询效率.

参考文献:

[1] I Stoica, R Morris, et al. Chord: A scalable peer-to-peer lookup service for internet applications [A]. ACM International Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications [C]. San Diego, USA: ACM Press, 2001. 149 – 160.

[2] S Ratnasamy, P Francis, et al. A scalable content-addressable network [A]. ACM International Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications [C]. San Diego, USA: ACM Press, 2001. 161 – 172.

[3] A Rowstron, P Druschel. Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems [J]. Lecture Notes in Computer Science, 2001, 2218 (2001): 329 – 350.

[4] Y Tang, S Zhou. LHT: A low-maintenance indexing scheme over DHTs [A]. IEEE International Conference on Distributed Computing Systems [C]. Beijing, China: IEEE Computer Society, 2008. 141 – 151.

[5] M Balazinska, H Balakrishnan, et al. INS/Twine: a scalable peer-to-peer architecture for intentional resource discovery peer-to-peer systems [J]. Lecture Notes in Computer Science, 2002, 2414(2002): 195 – 210.

[6] C Tang, Z Xu, et al. Peer-to-peer information retrieval using self-organizing semantic overlay Networks [A]. ACM International Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications [C]. Karlsruhe, Germany: ACM Press, 2003. 175 – 186.

[7] S Zhou, Z Zhang, et al. SIPPER: Selecting informative peers in structured P2P environment for content-based retrieval [A]. IEEE International Conference on Data Engineering [C]. Atlanta, USA: IEEE Computer Society, 2006. 161 – 161.

[8] Y Li, H V Jagadish, et al. SPRITE: A learning-based text retrieval system in DHT networks [A]. IEEE International Conference on Data Engineering [C]. Istanbul, Turkey: IEEE Computer Society, 2007. 1106 – 1115.

[9] T Pitoura, N Ntarmos, et al. Replication, load balancing and efficient range query processing in DHTs [J]. Lecture Notes in Computer Science, 2006, 3896(2006): 131 – 148.

[10] M Cai, M Frank, et al. MAAN: a multi-attribute addressable network for grid information services [J]. Grid Computing, 2004, 2(1): 3 – 14.

[11] A Bharambe, P Agrawal, et al. Mercury: supporting scalable multi-attribute range queries [J]. SIGCOMM Computer Communication Review, 2004, 34(4): 353 – 366.

[12] H Shen, Apon, et al. LORM: Supporting low-overhead P2P-based range-query and multi-attribute resource management in grids [A]. IEEE International Conference on Parallel and Distributed Systems [C]. Hsinchu, Taiwan: IEEE Computer Society, 2007. 1 – 8.

[13] C Schmidt, M Parashar. Flexible information discovery in decentralized distributed systems [A]. IEEE International Symposium on High Performance Distributed Computing [C]. Seattle, Washington, USA: IEEE Computer Society, 2003. 226

– 235.

[14] I Podnar, M Rajman, et al. Scalable peer-to-peer web retrieval with highly discriminative keys [A]. IEEE International Conference on Data Engineering [C]. Istanbul, Turkey: IEEE Computer Society, 2007. 1096 – 1105.

[15] M Datar, N Immorlica, et al. Locality-sensitive hashing scheme based on p-stable distributions [A]. ACM Symposium on Computational Geometry [C]. New York, USA: ACM Press, 2004. 253 – 262.

[16] H Shen, Z Li, et al. PIRD: P2P-based intelligent resource discovery in internet-based distributed systems [A]. IEEE International Conference on Distributed Computing Systems [C]. Beijing, China: IEEE Computer Society, 2008. 858 – 865.

[17] D Kempe, A Dobra, et al. Gossip-based computation of aggregate information [A]. IEEE Symposium on Foundations of Computer Science [C]. Cambridge, MA, USA: IEEE Computer Society, 2003. 482 – 491.

[18] Y Hua, B Xiao, et al. Bounded LSH for similarity search in peer-to-peer file systems [A]. IEEE International Conference on Parallel Processing [C]. Portland, Oregon, USA: IEEE Computer Society, 2008. 644 – 651.

[19] P Indyk. Stable distributions, pseudorandom generators, embeddings, and data stream computation [A]. IEEE Symposium on Foundations of Computer Science [C]. Redondo Beach, CA, USA: IEEE Computer Society, 2000. 482 – 491.

[20] P Indyk and R Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality [A]. ACM Symposium on Theory of Computing [C]. Dallas, Texas, USA: ACM Press, 1998. 604 – 613.

[21] M Jelasity, A Montresor, et al. PeerSim: A Peer-to-Peer Simulator [OL]. SourceForge project. http://peersim.sourceforge.net, 2009.

[22] W Meier. eXist: An open source native XML database [J]. Lecture Notes in Computer Science, 2003, 3841(2003): 189 – 200.

[23] Z Deng, D Feng, et al. Range query using learning-aware RPS in DHT-based peer-to-peer networks [A]. IEEE International Symposium on Cluster Computing and the Grid [C]. Shanghai, China: IEEE Computer Society, 2009. 180 – 187.

作者简介:

邓　泽　男,1978 年出生于湖北武汉,博士研究生,研究方向为广域网下的资源管理.
E-mail: deng_ze@163.com

冯　丹　女,1970 年,博士,教授,博士生导师,研究方向为计算机外存储系统、磁盘阵列、海量信息存储.

周　可　男,1974 年生于湖南湘潭,博士,教授,研究方向为网络存储、网络数据安全与服务、并行 I/O.

施　展　男,1976 年,博士研究生,讲师,研究方向广域网文件系统.