软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2011,22(9):2104−2120 [doi: 10.3724/SP.J.1001.2011.03878]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

利用多级局部性实现可扩展的无结构 P2P 搜索*

李治军+, 姜守旭, 李晓义

(哈尔滨工业大学 计算机科学与技术学院,黑龙江 哈尔滨 150001)

Exploiting Multi-Level Locality to Implement the Scalable Search in Unstructured P2P Network

LI Zhi-Jun+, JIANG Shou-Xu, LI Xiao-Yi

(School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China)

+ Corresponding author: E-mail: lizhijun_os@hit.edu.cn

Li ZJ, Jiang SX, Li XY. Exploiting multi-level locality to implement the scalable search in unstructured P2P network. *Journal of Software*, 2011,22(9):2104−2120. http://www.jos.org.cn/1000-9825/3878.htm

Abstract: Some statistical characteristic will emerge in unstructured P2P networks because of its large scale. Using above phenomena, an unstructured P2P overlay called Multi-Level Local Overlay, or ML^{2}O is presented in this paper. The mathematical control on the links can generate a topology with locality in multiple scale. Theoretical analyses show that the diameter of network and the average degree of peers are all $O(\log n)$ where n is the size of peers. Based on the multi-level locality, a recursive indexing mechanism is devised in this paper, in which the index for larger locality is build from the indexes of smaller localities Finally, an unstructured P2P search algorithm called Local Pervasion and Directed Search, or LPDS is presented in this paper. LPDS will collect information from local scope by executing local pervasion. Moreover, a part of the index tree can be achieved from the collected information. LPDS can find the next hop approaching the destination in the partial index tree. Theoretical analyses show that the expectation of average search hops and communication loads produced by LPDS are all $O(\log n)$. Experimental results illustrate the scalability of LPDS on ML^{2}O is close to structured P2P search and its robustness is close to unstructured P2P search.

Key words: unstructured P2P network; multi-level locality; index mechanism; local pervasion and directed search

摘 要: 无结构 P2P 网络拓扑随着规模的增大会出现一定的统计特性,充分应用该现象提出了一种多级局部覆盖网络(multi-level local overlay,简称 ML^{2}O)的无结构 P2P 覆盖网,对 ML^{2}O 中节点间的连接进行恰当的数学控制后,就能使产生的拓扑具有从微观到宏观的多个粒度上的局部性.理论分析表明,ML^{2}O 的网络直径和节点平均度都是网络规模 n 的对数,为其上建立可扩展的无结构 P2P 搜索奠定了基础.给出了应用 ML^{2}O 多粒度局部特性的索引机制:首先以局部为单位建立信息索引;然后在局部索引的基础上建立更大粒度局部的索引,从而形成一棵索引树;最后提出了一种局部渗透定向搜索算法(local pervasion and directed search,简称 LPDS).LPDS 用局部渗透收集到的信息建立部分索引树,并在树上找到更接近搜索目标的下一跳.理论分析表明,LPDS 搜索算法的平均搜索跳数和通信

* 基金项目: 国家自然科学基金(60803148, 60973124)
收稿时间: 2009-09-19; 定稿时间: 2010-04-27

负载都是 $O(\log n)$.模拟实验结果表明,ML^2O 上 LPDS 的可扩展性接近结构化 P2P 搜索,其鲁棒性接近无结构 P2P 搜索.

关键词: 无结构 P2P 网络;多级局部性;索引机制;局部渗透定向搜索

中图法分类号: TP393 **文献标识码:** A

搜索是 P2P 信息共享系统中的研究重点,这是由于搜索是直接影响信息共享服务质量(如搜索应答的时延)和 P2P 系统性能(如搜索产生的通信负载)的关键因素[1].在无结构 P2P 搜索中,发起查询的节点选取部分或全部邻居节点发送查询数据包,收到查询的节点如果存有文件则应答查询节点,否则选取部分或全部邻居节点转发查询数据包.这个简单的搜索过程使无结构 P2P 搜索的鲁棒性很好:动态变化对搜索几乎不产生影响,搜索可在任何网络拓扑上顺利工作等.但无结构 P2P 搜索的高鲁棒性的代价是搜索效率低、可扩展性差.

泛洪搜索(flooding)是将查询包广播到网络上所有节点的一种搜索方法,该方法造成的通信负载显然要远大于节点个数 n,即使能够提供搜索应答的节点离查询请求节点很近.常用扩展环状搜索(expanding ring)来解决该问题,此时,用 TTL 控制搜索逐层向外广播($TTL=n$ 没得到搜索应答,设 TTL 为 $n+1$ 继续广播),直到收到应答为止.因此,扩展环状搜索实际上就是一个从 q 出发的半径为 $D(q,d)$(请求节点到目标节点的距离)的泛洪,单副本情况下,$D(q,d)$的均值为网络半径 R 的一半,因此,搜索的通信负载的下界就是和节点 q 相距 $R/2$ 以内的节点个数.这一个数的均值应至少为 n 的多项式,因为对于任何一对节点 p,q,在 p 和 q 处同时发起的半径为 $R/2$ 的泛洪一定搜遍 n 个节点,这表示网络中至少有一半节点满足“距其 $R/2$ 以内的节点个数不小于 $n/2$”.表明了扩展环状搜索的通信复杂度仍是 n 的多项式.

随机游走搜索是另一种不广播的搜索:节点在收到查询数据包时随机找一个邻居节点转发,此时,在第 k 跳搜索成功的概率是一个几何随机变量,k 的数学期望(见公式(1))表明,搜索的通信负载和用户延迟也都是 n 的多项式,随机游走只是将搜索的通信负载平摊在时间轴上,尽量避免节点在某段时间因处理的查询数据包过量而拥塞.

总的来说,无结构 P2P 上基于泛洪(包括限制泛洪)和随机游走的搜索,其通信负载都是 n 的多项式,可扩展性差[2−4].而本文旨在研究一种可扩展的无结构 P2P 搜索机制:利用局部性来提高无结构 P2P 上的搜索效率,其核心思想是通过对局部性的数学控制来实现一种称其为多级局部覆盖网络(multi-level locality overlay,简称 ML^2O)的多粒度局部性结构,然后在多级局部覆盖网络上部署合适的索引来实现可扩展的无结构 P2P 搜索.

本文第 1 节对无结构 P2P 网络上的搜索机制进行全面的总结,并对本文进行对比分析.第 2 节提出多级局部性概念和多级局部覆盖网络的结构、性质和构造算法.第 3 节给出在 ML^2O 上建立的索引.以该索引结构为基础,第 4 节提出相应的无结构 P2P 搜索算法,并对该搜索的可扩展性进行理论分析.第 5 节在实际 P2P 工作环境中对本文提出的搜索算法进行了大量模拟实验,分析该算法在实际环境下的效果.

1 相关工作

引入副本提高 P2P 搜索效率是一类常见方法,其思想是,在适当节点上备份文档来提高搜索效率[4−6].文献[4]证明,基于 random walk 查询路径的备份机制是均匀抽样,并据此分析了该方法对搜索效率的影响.如果文档 d 的副本在网络中均匀分布时,基于 random walk 的无结构搜索算法能在第 k 跳搜索到文档 d 的概率 $Pr(k,d)$为

$$Pr(k,d)=\frac{r(d)}{n}\left(1-\frac{r(d)}{n}\right)^{k-1} \tag{1}$$

其中,$r(d)$是文档 d 在网络中的副本个数.公式(1)表明,搜索成功所需的跳数是一个以 $r(d)/n$ 为参数的几何随机变量,所以平均搜索跳数为 $n/r(d)$,只要 $r(d)$没有超过 $n/\log(n)$,通信负载仍是 $O(p(n))$($p(n)$表示 n 的多项式).由于空间复杂度和副本一致性等问题,$r(d)\geqslant n/\log(n)$时产生的大量备份对于实际 P2P 环境是不可行的.文献[5]对无结构 P2P 网络中的多种文档备份策略进行了详细分析,但其基础仍是公式(1)的概率值,也没有从根本上给出可扩展的无结构 P2P 搜索.BubbleStorm[6]提出了一种有趣的方案:不仅备份文档数据,还将查询备份,即在路经的节

点上存储查询.此时,搜索成功就是查询备份和文档备份在某节点上出现碰撞,查询 q 和文档 d 出现碰撞的概率 $Pr(q,d)$为

$$Pr(q,d)=1-\left(1-\frac{r(d)\times r(q)}{n^2}\right)^n\approx 1-\mathrm{e}^{-\frac{r(d)\times r(q)}{n}} \tag{2}$$

其中,$r(q)$是查询 q 在网络中的副本个数.

要使 BubbleStorm 的搜索成功率大于某个固定的数值,需要使 $r(d)\times r(q)\geqslant cn$.与上面的分析结果一样,当文档副本个数 $r(d)$的规模不是很大时,仍有 $r(q)=O(p(n))$.

引入索引,使查询包在路由过程中尽快找到应答节点,是提高无结构 P2P 搜索效率的又一常见方法[7−9].GIA[7]中的每个节点都存有其一跳内邻居节点上存放数据的索引,而由于 P2P 网络的异构性,某些能力较高的节点通常拥有较多的邻居节点.所以,P2P 搜索时如果偏向于随机游走这些节点,就能够有效减少搜索的通信负载.从理论分析角度,如果查询数据包到了 GIA 中的某个节点,则相当于查询了 GIA 中以该节点为中心的一跳范围内的节点集合.当文档没有备份时,搜索的通信负载的下界时满足公式(3)的 l:

$$\sum_{i=p_1}^{p_l}\lambda(i)\geqslant n \tag{3}$$

其中,$p_1,...,p_l$是搜索路径上的节点,$\lambda(i)$为节点 i 的度.如果没有采用有偏的随机游走,$l=n/\Lambda$(Λ是节点平均度);如果偏向于游走$\lambda(i)$较大的节点,则 $l=n/\Lambda'$,Λ'是度较大节点集合的平均度.由于无结构 P2P 网络是稀疏网络,Λ必远小于 $p(n)$;而对于Λ',必有 n/Λ'大于 P2P 网络最小连通支配集的规模.对于无结构 P2P 网络,该值仍然是 $p(n)$.这一点可从文献[8]的理论分析结果得到验证.所以,GIA 搜索的理论可扩展性仍是 $p(n)$.尽管文献[7]的实验结果表明 GIA 能够有效提高搜索效率,但这些结果是在大量假设下得出的,如假设“Most queries are for hay,not needles”[7].在 GIA 基础上,自然想到的方法是建立多跳范围内的索引,文献[8]给出的就是这样一类工作.该文以 two-hop index 机制为基础,再辅以“网络拓扑控制”和“限制在超级节点上的随机游走”将 GIA 的思想形成一个完整的体系.但搜索的通信负载显然只是将以公式(5)中的$\lambda(i)$变为$\lambda^2(i)$(节点 i 两跳内的邻居个数),搜索的理论可扩展性仍然是 $p(n)$.文献[8]中的大量理论分析给出了类似的结论.

Routing Indices(简称 RI)[9]也是一种多跳索引机制,但与文献[7,8]存在的一个显著不同是,RI 引入了索引聚合概念,$goodness(N_i,Q)$根据邻居节点 N_i 处聚合的 RI 评价 N_i 对查询 Q 的适应度,RI 的搜索下一跳选择 *goodness* 最大的邻居节点.RI 的搜索负载取决于索引定义、索引聚合、网络拓扑等诸多因素,文献[9]没有对该复杂度进行理论分析.理想情况下,如果节点上维护的 RI 使 *goodness* 总能从邻居节点中选出正确的下一跳节点,RI 的搜索负载就是网络直径,此时是可扩展的.但这个可扩展的无结构 P2P 搜索需要任意节点上的 RI 聚合了网络上全部节点的索引信息,此时,建立和维护 RI 的复杂度就变成了 $O(n)$.因此,文献[9]仍不可扩展,但其给出的索引及索引聚合对提供可扩展的无结构 P2P 搜索有重要的借鉴意义,本文提出的无结构 P2P 搜索算法就应用了该思想.

应用局部性(locality)提高 P2P 搜索性能是目前常见的研究[10−16],且从这类研究中可以看出:在 P2P 搜索中有多种局部性,如时间局部性、空间局部性、兴趣局部性、语义局部性可以被用来提高 P2P 搜索的效率,表明局部性是 P2P 搜索的一个典型特征,且大量结果也表明了应用局部性显著地提高了无结构 P2P 上的搜索效率.这也是本文研究的动机,本文在总结和分析现有基于局部性进行 P2P 搜索研究不足的基础上提出了多级局部性概念,并据此给出了一种理论证明可扩展的、在实际复杂环境中也可良好工作的无结构 P2P 搜索算法.

文献[10]用大量的模拟实验和从实际系统采集数据集上的仿真实验都表明,基于兴趣的局部性可以提高搜索效率.有趣的是,文献[10]只给出了引入 *interest-based shortcuts* 这样一种简单的机制就能获得良好的效果,使得搜索引起的通信负载减少 50%以上.这个有趣的结论表明了,在局部性基础上引入一条简单的机制就能取得一定的效果;那么如果引入的是基于局部性建立的完整框架,可能会从本质上提高无结构 P2P 搜索的可扩展性(但对文献[10]提出的机制,其搜索可扩展性仍是 $O(n)$).Foreseer[11]也是一个利用局部性提高搜索效率的系统,它利用 P2P 上空间局部性(建立了称为 *Neighbor Overlay* 的覆盖网络来利用这类局部性)和时间局部性(对应的覆盖网络是 *Friend Overlay*)来提高无结构 P2P 的搜索效率,模拟实验结果表明,Foreseer 较现有的无结构 P2P 搜

索算法,如泛洪、索引等机制,可减少 50%~90%的应答时间和通信负载.但 Foreseer 系统从本质上和文献[10]的捷径一样,只不过是引入两类捷径而已:*Neighbor Overlay* 对应一类捷径,*Friend Overlay* 对应另一类捷径.所以,Foreseer 搜索可扩展性的理论值仍为 $O(n)$,没有从根本上解决无结构 P2P 搜索的可扩展性问题.

应用节点存储内容之间的语义关系来缩小 P2P 搜索空间、提高搜索效率也是局部性在 P2P 搜索上的一类典型应用[12−14].文献[12]将 ontology tree 转化为一棵 partition tree,用该树中的节点负责对应的那部分语义空间,并将这些节点映射到 HyperCup 中进行 P2P 信息发布和搜索.所以,该文将此结构称为 ontology-based DHT (ODHT).文献[12]的理论分析表明,搜索的通信负载是 $O(\sqrt{2^D})$(其中,D 是 ontology tree 的高).如果以节点个数为问题的规模,ODHT 提供了可扩展的搜索,但如果以 ontology tree 中的节点个数为问题规模,ODHT 的搜索仍是不可扩展的.这样的结果代表了此类研究的共性结果[12,13]:将语义和 DHT 组合增加了 P2P 搜索的能力(支持更复杂的搜索);DHT 的引入使得搜索对于节点规模 n 是可扩展的.但同时引入节点 churn 导致 DHT 的维护问题[15];语义结构的引入造成另外的复杂性.如 ODHT 中给出的 $O(\sqrt{2^D})$的搜索负载[12],当语义网络的规模很大时,又引入了其他可扩展性问题.相比 ODHT,SemreX[14]给出的研究是在无结构 P2P 网络上应用语义局部性提高搜索效率.SemreX 使用 Latent Semantic Indexing[16]描述文档并计算文档间的距离,将语义相似度高的节点局部聚类在一起,并类似 Foreseer[11]引入长距离捷径连接(long-range connections).在这样的 semantic overlay 上,SemreX 给出了基于相似度的启发式搜索算法,使大多数搜索能够在局部完成,提高搜索效率.同样的,SemreX 也代表了一类研究,如文献[17]提出的 associative overlays 使用 guide rule 来定义一个语义相似的节点集合(如存有某个歌唱家的音乐文件的节点集),每个节点优先将搜索转发那些更可能应答查询的邻居节点.显然,类 SemreX 的这类研究和文献[10,11]的研究在本质上一样,在局部性假设基础上的模拟实验结果会大幅降低通信负载,但在理论上仍然不能提供可扩展的无结构 P2P 搜索.

总之,上述无结构 P2P 网络的研究中都没有提出理论证明可扩展的 P2P 搜索算法.相比无结构 P2P 网络,结构化 P2P 网络[18]基于严格的分布式结构通常都能实现一个可扩展的 P2P 搜索,如 Chord[18]通过每个节点维护的 finger 表和环上后继节点表实现了 $O(\log n)$的搜索跳数和搜索通信负载.同样的,其他结构化 P2P 网络,如 Tapestry[19]和 CAN[20]也都能获得 $O(\log n)$的 P2P 搜索扩展性.但由于结构化 P2P 网络的拓扑过于脆弱,又由于查询的转发和网络拓扑紧密关联,使得该网络随着节点的动态加入和离开而产生了很大的维护开销[19].除基于 DHT 的结构化 P2P 网络以外,文献[21,22]给出了基于树的多级网络拓扑.文献[21]中的 HAT 是基于结点能力等级构建的树结构,文献[22]中 HASN 基于节点间的物理距离形成了普通节点 peer 和分级代理节点 HA 组成的树结构(树的叶子节点是 peer,树的内部节点是 HA).由于采用树结构,HAT 和 HASN 都表现出良好的搜索性能(其搜索扩展性为树高,通常为 $O(\log n)$).但这两个方案都需要显式地建立并维护树形拓扑,系统的鲁棒性差、维护代价高,存在和结构化 P2P 网络一样的问题[15].而在当前现实 P2P 环境中,节点不断动态变化是其典型特征[23],因此,目前的 P2P 应用更倾向于建立在无结构 P2P 网络上[2,6,7],如 Gnutella,Kazaa,BitTorrent 等.在这样的背景下,给出一种搜索跳数和通信负载复杂度接近 $O(\log n)$的可扩展无结构 P2P 搜索方法,就成为 P2P 研究的一个开放问题.本文提出了一种在拓扑结构和索引数据一致时搜索复杂度为 $O(\log^2 n)$的无结构 P2P 搜索算法,而且该方法在拓扑维护和索引建立时的额外通信开销也是可扩展的,为 $O(\log n)$.

对于能够从理论上证明可扩展的无结构 P2P 搜索,需要提到 Stratis Ioannidis 等人的研究成果:作者首次给出了能理论证明可扩展的并且是可靠的无结构 P2P 搜索算法[24].文献[24]的搜索算法以随机游走为基础,采用副本备份机制,加入了文档不存在证据(evidence of absence)机制.在这 3 个机制下,文献[24]证明了:(1) 对于任何的查询文档 d 的节点数量和 d 的副本数量,每个节点在任何时刻处理的针对 d 的搜索通信负载为 $O(1)$;(2) 只要文档 d 的副本数量为 $\omega(1/n)$,则搜索成功率随着 n 增大趋近 1.结果(2)很好地证明了该搜索算法的可靠性,但结果(1)只表明节点在任意时刻的通信负载是可扩展的,但这样的可扩展性实际上是由于随机游走将大量的搜索通信负载平摊在时间轴上获得的.所以,文献[24]给出的可扩展搜索在用户延迟(搜索跳数)上仍是不可扩展的.相比文献[24]而言,本文给出的无结构 P2P 搜索算法在通信负载和搜索跳数上都是可证明扩展的.

2 多级局部覆盖网络

2.1 多级局部性

搜索的局部性表明,P2P 节点更有可能从邻近的节点那里得到查询应答.所以,P2P 节点应尽量建立和邻近节点的连接,然后让搜索算法优先查找这些邻近节点就能提高搜索效率[10−16].因此,局部感知无结构 P2P 搜索首先需要定义节点间的邻近关系和节点间的连接.节点的邻近体现为节点间的距离更短(距离的定义取决于邻近的含义,而此处的邻近有广泛的含义,如表示节点间的物理通信延迟小、节点存放内容的语义相似、节点的查询兴趣相同等[10−16]),因此,基于局部性的 P2P 节点连接是由节点间距离决定的:一个直观的想法是,节点间的距离越小,节点间建立连接的可能性就越大.

这样,一个直观想法导致的直接结果就是:距离较近的若干节点会紧密相连,这是由于节点 p 和节点 q 的距离较小,节点 q 和节点 o 的距离较小,则节点 p 和节点 o 的距离也较小,节点间的距离定义通常都满足三角不等式,如物理延迟距离定义或语义相似度距离定义等.距离较近的若干节点的互连,实际上就是节点间的连接在局部上出现汇聚的现象,这就是常说的“簇”结构[10−16].关于簇结构的研究很多,但常见的研究都是基于簇现象提出或改进了某些搜索算法,大都没再深入分析“簇”结构的细节[10−16].实际上,簇刻画的是在节点粒度上的节点间连接出现的汇聚现象,那么在更大的粒度上的连接关系(即簇与簇之间)是否也具有类似的性质,目前还很少被研究和使用.以节点为单位的汇聚会形成小规模的簇,以小规模簇为单位会汇聚出更大规模的簇,这就是被本文称为“多级局部性”的结构特征,本文正是使用这一特征来刻画大规模无结构 P2P 网络中隐含的结构.

如果一个节点 o 的邻居节点和节点 o 邻近,那么这些邻居节点之间也会邻近,由连接的局部性决定了这些邻居节点也会以很高的概率互连,此时的节点 p 的簇系数 $C_e(p)$就会较大.所以,可用簇系数来描述网络是否出现局部性特征.

定义 1(节点 p 簇系数 $C_e(p)$). 对于 P2P 网络中的任一节点 p,其节点簇系数 $C_e(p)$定义为

$$C_e(p)=\frac{|E\cap(N(p)\times N(p))|}{|N(p)|\times(|N(p)|-1)} \tag{4}$$

其中,E 是 P2P 网络的边集,$N(p)$是节点 p 的邻居节点集合,$N(p)\times N(p)$是 p 所有邻居节点的笛卡尔积.

较大的节点簇系数表示 P2P 网络在节点 p 处有的局部汇聚成簇的趋势.公式(4)定义的是节点簇系数,可将公式(4)推广到节点集合的簇系数的定义.

定义 2(节点集合 S 簇系数 $C_s(S)$). 对于 P2P 网络中的任意节点集合 S,其节点集合簇系数 $C_e(S)$定义为

$$C_s(S)=\frac{|E\cap(S\times S)|}{|S|\times(|S|-1)} \tag{5}$$

根据公式(4)和公式(5)可知,节点簇系数可用节点集合簇系数表示,即 $C_e(p)=C_s(N(p))$.这表明节点集合簇系数 $C_s(S)$不仅可以表示 P2P 网络在节点 p 处的局部汇聚现象,而且在局部性刻画方面具备更强的能力.本文给出的多级局部覆盖网络正是用 $C_s(S)$来定义的.

2.2 多级局部覆盖网络

定义 3(多级局部覆盖网络 $\mathbf{ML^2O}$). 如果某 P2P 的覆盖网络 $G=(P,E)$满足如下性质,则称该覆盖网络为多级局部覆盖网络:

(a) $\forall p\in P(G),\exists C\subseteq P(G)$和某阈值 T 且 $C_S(C)\geqslant T$,有 $p\in C$;

(b) 条件(a)表明,每个节点至少属于某簇 C.所以,存在由 $P(G)$的子集组成的集族$\{C_1^{(1)},C_2^{(1)},\ldots,C_u^{(1)}\}$满足条件

$$P(G)=\bigcup_{i=1}^{u}C_i^{(1)} \tag{6}$$

(c) 在条件(b)的基础上定义 G 导出的网络 $G^{(2)}(G=G^{(1)})$,其中,$G^{(2)}$的节点集合为 $C_i^{(1)}$,即将 $C_i^{(1)}$ 整体作为一个节点.$G^{(2)}$中的边定义如下:对任何两个节点 $C_i^{(1)}$ 和 $C_j^{(1)}$,如果存在若干对 p 和 q(序对的具体数量可

根据实际问题的需要来定义,本文定义为簇的平均规模),$p\in C_i^{(1)}$ 和 $q\in C_j^{(1)}$ 有$(p,q)\in E(G)$,则

$$(C_i^{(1)},C_j^{(1)})\in E(G^{(2)}).$$

(d) 对于导出图 $G^{(2)}$,$G^{(2)}$满足条件(a);

(e) 按照同样的方法在 $G^{(2)}$定义导出图 $G^{(3)}$,类似地定义 $G^{(4)}$,…,其中,所有的导出图 $G^{(i)}$都满足条件(a);

(f) 当 $G^{(L)}$是包含一个节点的图时,导出过程结束.

定义 3 表明,多级局部覆盖网络在从微观到宏观的多个角度上都具有局部凝聚的性质,反映了拓扑局部性的多粒度特性,该特性可用图 1 形象描述.

定理 1. 对于包含 n 个节点的多级局部覆盖网络,平均情况下有 $L=O(\log n)$.

证明:由定义 3 可知,在导出 $G^{(l+1)}$时要从 $G^{(l)}$¬¬的节点集合选出一些子集作为 $G^{(l+1)}$的节点.如果构成 $G^{(l+1)}$的节点集族 $\{C_1^{(l)},C_2^{(l)},...,C_u^{(l)}\}$ 两两互不相交,再由公式(7)可知,$G^{(l)}$的节点个数=$|C_1^{(l)}|+|C_2^{(l)}|+...+|C_u^{(l)}|=\hat{C}\times G^{(l+1)}$的节点个数,其中,$\hat{C}$ 为簇大小的平均值.这样,$G^{(l+1)}$的节点个数就是 $G^{(l)}$节点个数的 $1/\hat{C}$.由于对 $1\leqslant l\leqslant L$ 该关系都成立,而且 $G^{(L)}$的节点个数为 1,所以有 $L=O(\log_{\hat{C}}n)=O(\log n)$.

但在实际的 P2P 覆盖网络中,$C_1^{(l)},C_2^{(l)},...,C_u^{(l)}$ $(1\leqslant l<L)$常会出现相交的情况.出现在多个簇中的节点通常都是网络中的超级节点,而这样的节点在每个簇中只占很少一部分[25].图 2 给出了至多只有一个超级节点同时出现在多个簇中时,簇和这些超级节点间的关系.可对 $G^{(i)}$中的节点定义计数值:节点每出现在一个新的簇中时计数值就增加 1.显然,除这些出现在多个簇中的超级节点以外,其他节点的计数为 1.对于超级节点可以通过枚举其出现的簇得到计数值,如在图 2 中该枚举结果为$\{\{C_i,C_j,C_k\},\{C_l\},\{C_m,C_n\}\}$.由于至多只有一个超级节点同时出现在多个簇中,所以这个枚举结果中的各个子集互不相交,因此,这些超级节点获得的计数值之和不大于 $G^{(i)}$的簇的个数.按此种计数方法,$G^{(i)}$中所有节点的计数值之和至多为 n_i+n_{i+1},其中,n_i 是 $G^{(i)}$中的节点个数,而 $G^{(i)}$的簇就是 $G^{(i+1)}$的节点,所以 n_{i+1} 就是 $G^{(i)}$的簇的个数.

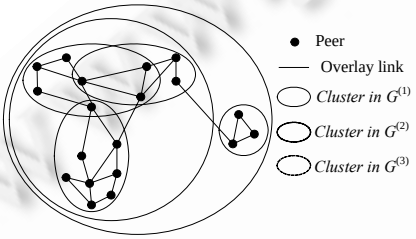

Fig.1 An instance of multi-level locality overlay

图 1 一个多级局部覆盖网络实例

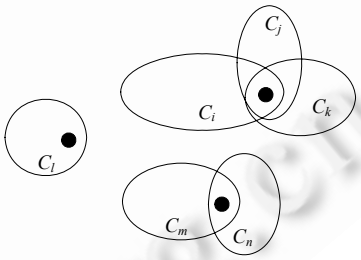

Fig.2 One super-peer appear in multiple clusters

图 2 一个超级节点出现在多个簇中情形

可以按另一种简单方式完成上面的计数:将 $G^{(i)}$中所有簇的节点个数相加,即$|C_1^{(i)}|+|C_2^{(i)}|+...+|C_u^{(i)}|$.联立两个节点就有 $n_{i+1}\times\hat{C}\leqslant n_i+n_{i+1}$,因此

$$n_L\leqslant\frac{1}{\hat{C}-1}\times n_{L-1}\leqslant\left(\frac{1}{\hat{C}-1}\right)^2\times n_{L-2}\leqslant...\leqslant\left(\frac{1}{\hat{C}-1}\right)^{L-1}\times n_1.$$

由于 $n_L=1$,所以$(\hat{C}-1)^{L-1}\leqslant n$,有 $L=O(\log_{\hat{C}-1}n+1)=O(\log n)$.

对于有多个(如 s 个)超级节点同时出现在多个簇中的情形,仍使用上述的计数定义和枚举方法,有 $n_{i+1}\times\hat{C}\leqslant n_i+s\times n_{i+1}$.$s$ 通常比 $\hat{C}$ 要小很多,显然仍有 $L=O(\log n)$的结论. □

2.3 ML^{2}O的建立

根据 ML^{2}O 具有的局部汇聚特性,建立 ML^{2}O 的直观想法是,两个节点距离越近,其建立连接的概率就越大.为实现多粒度局部汇聚,本文定义节点 p 和 q 的簇距离 $\kappa(p,q)$,节点 p 和 q 建立连接的概率是簇距离 $\kappa(p,q)$的函数.

定义 4(簇距离 $\kappa(p,q)$). 在多级局部覆盖网络上,节点 p 和 q 的簇距离 $\kappa(p,q)$定义为

$$\kappa(p,q)=\min\{i\mid(\exists C_l^{i-1}\in P(G^{(i)}))p\in C_l^{i-1}\wedge q\in C_l^{i-1}\} \tag{7}$$

随着观察的粒度增大,节点 p 和 q 最终定位于同一簇中(因为观察全部节点时 p 和 q 显然位于同一簇).公式(7)说明,簇距离 $\kappa(p,q)$ 就是能在一个簇中发现节点 p 和 q 的最小观察粒度.

定理 2. 用公式(8)作为节点 p 和 q 建立连接的概率($Pr(Connect(p,q))$就是 p 和 q 建立连接的概率)时

$$Pr(Connect(p,q))=\frac{1}{\hat{C}^{\kappa(p,q)-1}} \tag{8}$$

则形成的覆盖网络是多级局部覆盖网络 ML^{2}O.

证明:对于 G 中任何一节点 p 和满足 $\kappa(p,q)=1$ 的节点 q,由公式(10)可知,概率 $Pr(Connect(p,q))$ 接近于 1,可认为节点 p 连接 q.再根据定义 4,簇距离对应的二元关系具有对称性 $\kappa(p,q)=\kappa(q,p)$,节点 q 也连接节点 p.因此,满足 $\kappa(p,q)=1$ 的所有节点对以很高的概率互连,形成一个簇系数接近于 1 的簇.该簇是满足定义 3 中条件(a)的 C.

簇距离 κ 不仅是一个对称关系,显然,κ 也满足自反性和传递性.所以,κ 是一个等价关系.可用 κ 对 G 进行划分,其划分结果就是 $G^{(1)}$ 的簇(也就是 $G^{(2)}$ 的节点集).对于其中任何一个簇 $C_i^{(1)}=\{p_1,p_2,\ldots,p_{\hat{C}}\}$ 和 $C_j^{(1)}=\{q_1,q_2,\ldots,q_{\hat{C}}\}$(为简化分析,假定 $C_i^{(1)}$ 和 $C_j^{(1)}$ 的规模都是 $\hat{C}$,一般情况下的期望分析结果是一致的),如果 $\kappa(p_j,q_k)|_{1\leqslant j,k\leqslant\hat{C}}=2$,由公式(10)得知 $Pr(Connect(p,q))=1/\hat{C}$.此时,节点序对集合 $\{(p,q)|p\in C_i^{(1)},\ q\in C_j^{(1)},\ (p,q)\in E(G)\}$ 的基数的平均值为 $\hat{C}^2/\hat{C}=\hat{C}$,由定义 3 得知,$C_i^{(1)}$ 和 $C_j^{(1)}$ 在 $G^{(2)}$ 以很高的概率存在连接.由于 $G^{(1)}$ 中和节点 p 满足性质 $\kappa(p,q)=2$ 的节点个数的平均值为 $\hat{C}^2$,因此 $G^{(2)}$ 中和 $C_i^{(1)}$ 以接近于 1 的概率互连的簇个数的平均值为 $\hat{C}^2/\hat{C}$.这说明对于 $G^{(2)}$ 中的任何一个节点 $C_i^{(1)}$,存在一个包含 $C_i^{(1)}$ 的规模为 $\hat{C}$、簇系数接近 1 的节点集合,满足定义 3 中的条件(d).

用同样的思路可以证明定义 3 中的条件(e).

由于 $\hat{C}>1$,所以上述划分过程必然会使 $G^{(i+1)}$ 的节点个数小于 $G^{(i)}$ 的节点个数.因此,上述过程必然会使最终的 $G^{(L)}$ 只包含一个节点,满足定义 3 中的条件(f). □

定理 3. 对于按照公式(8)构造的包含 n 个节点的多级局部覆盖网络 G,$(\forall p)p\in P(G)$ 都有 $|N(p)|=O(\log n)$.

证明:对于 $\forall p\in P(G)$,定义和节点 p 距离为 $i(1\leqslant i\leqslant L)$ 的节点集合为 $N^{(i)}(p)$,则 $N(p)=\bigcup_{i=1}^{L}N^{(i)}(p)$.

由于 $|N^{(i)}(p)|=|\{q|\kappa(p,q)=i\}|/\hat{C}^{(i-1)}$,其中,$|\{q|\kappa(p,q)=i\}|$ 是和节点 p 的簇距离为 i 的节点个数,$1/\hat{C}^{(i-1)}$ 是 $\{q|\kappa(p,q)=i\}$ 中的节点成为 p 的邻居的概率),显然有 $|\{q|\kappa(p,q)=i\}|=\hat{C}^i$,所以 $|N^{(i)}(p)|=\hat{C}$.

$$|N(p)|=\sum_{i=1}^{L}\hat{C}=\hat{C}\times L=O(\log n). \qquad \square$$

定理 1 和定理 3 表明,多级局部覆盖网络 ML^{2}O 具有类似于结构化 P2P 网络的拓扑性质:具有可扩展的节点度和网络半径,这为在 ML^{2}O 上实现可扩展的无结构 P2P 搜索奠定了基础.定理 2 表明,建立 ML^{2}O 的关键是计算两个节点的簇距离 $\kappa(p,q)$.但在计算 $\kappa(p,q)$ 时不能使用定义 4 中的 $G^{(i)}$,因为通过 $\kappa(p,q)$ 才建立了 ML^{2}O,所以在计算 $\kappa(p,q)$ 时不能使用 ML^{2}O 中的性质,而应该设计满足或近似满足定义 4、易计算的距离函数作为簇距离.

本文根据节点存放信息的语义相似度来计算 $\kappa(p,q)$.在描述文档信息的语义时,本文使用 Web Ontology Language(OWL)[26].图 3 的 ACM Topic ontology 描述了计算机领域语义概念间的一个层次关系(通过关系 IS-A 刻画)[27],在这个层次关系上,可以直观地定义父概念(super-topic)和子概念(sub-topic).在图 3 中,计算概念 A 和概念 B 的相似度时,从 A 和 B 出发的向上遍历过程中一定会找到一个最紧的共同概念 $\Delta(A,B)$(如 A 是图 3 中的,B 是图 3 中的,$\Delta(A,B)$ 是图 3 中的),可以用概念 $\Delta(A,B)$ 到概念 A 和概念 B 的较长的一条路径的路径长度作为定义 4 的 $\kappa(p,q)$ 的近似值.由于概念层次拓扑的树形结构和 ML^{2}O 的递归结构非常相似,所以用这个路径长度作为簇距离的近似很合适.

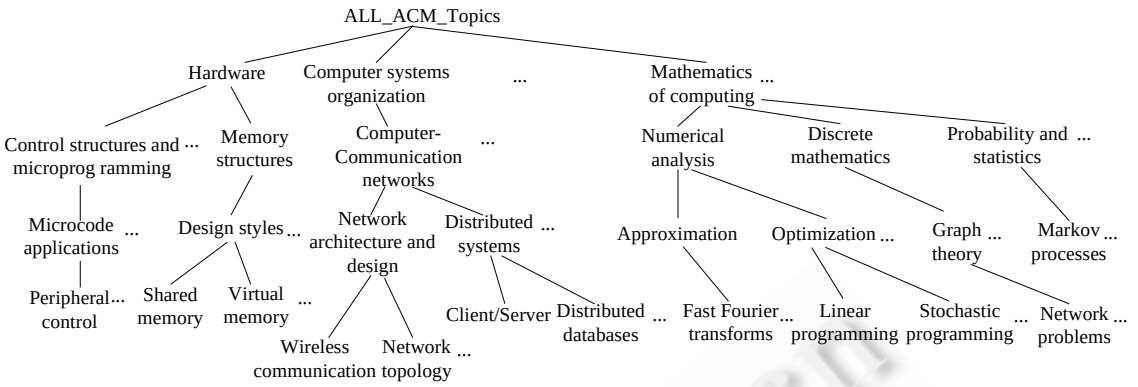

Fig.3 The structure of ACM topic

图 3 ACM 主题的结构

以上给出的是只考虑每个节点存放一种概念信息的情况,即给出了如何计算两个概念 A 和概念 B 的 $\kappa(A,B)$ 值,而一个实际的 P2P 节点上存放的信息是包含概念层次拓扑中的多个概念的信息集合,即 $\Gamma(p)=\{A_1,A_2,...,A_m\}$, $\Gamma(q)=\{B_1,B_2,...,B_{m'}\}$.此时,两个节点的 $\kappa(p,q)$ 值可在两个概念的 $\kappa(A,B)$ 值和节点对各概念的兴趣基础求加权平均

$$\kappa(p,q)=\sum_{i=1}^{|\Gamma(p)||\Gamma(q)|}\sum_{j=1}[\kappa(A_i,B_j)\times I(p,A_i)\times I(q,B_j)] \tag{9}$$

其中,$I(p,A_i)$是节点 p 处存放包含概念 A_i 的文档个数在存放文档总数中所占比例,反映节点 p 对概念 A_i 的兴趣.

3 多级局部覆盖网上的索引机制

需要对多级局部覆盖网络的节点处存放的信息建立索引,否则仅依据 ML^{2}O 的拓扑结构信息在处理 P2P 搜索请求时仍然只能进行盲目的转发,无法实现可扩展的无结构 P2P 搜索.ML^{2}O 的核心思想是形成局部汇聚的节点簇,所以为这些簇建立索引对 ML^{2}O 而言很自然.另一方面,ML^{2}O 中的局部簇是多粒度的,因此也就需要给各个粒度上的簇建索引.

图 4 是本文为 ML^{2}O 设计的一种索引结构,该索引以上文提到的语义概念和概念的运算 Δ 为基础:

(1) 给 $G^{(1)}$的节点建立的索引是该节点上存储信息的语义概念,如节点 p_1 处存放的语义概念为 O_1,则建立的索引项为$\langle p_1,O_1\rangle$(为记号统一,该索引项记为$\langle p_1^{(1)},O_1\rangle$);

(2) 当发现节点 $p_1,p_2,\ldots,p_m$ 在 $G^{(1)}$中形成簇时,引入记号 $p^{(2)}=(p_1,p_2,\ldots,p_m)$并建立索引项
$$\langle p^{(2)},\Delta(O_1,O_2,\ldots,O_m)\rangle.$$

(3) 同样地,当发现某些 $p^{(2)}$类节点按 ML^{2}O 定义在 $G^{(2)}$中形成簇时,引入记号 $p^{(3)}=(p_1^{(2)},p_2^{(2)},...,p_m^{(2)})$ 并建立索引项$\langle p^{(3)},\Delta(O_1,O_2,\ldots,O_m,O_{m+1},\ldots,O_{m+n},\ldots)\rangle$,其中,$O_1,O_2,\ldots,O_m$ 是节点 $p_1^{(2)}$ 对应的簇中节点上存放的语义概念,$O_{m+1},\ldots,O_{m+n}$ 是节点 $p_2^{(2)}$ 对应的簇中节点上存放的语义概念,等等;

(4) 依此类推可建立形如$\langle p^{(4)},O_x\rangle,\langle p^{(5)},O_y\rangle,\ldots,\langle p^{(L)},O_z\rangle$的索引项.

由于这些项具有明显的包含关系,可以自然地将其组织为如图 5 所示的树,根据其结构本文称其为 *Cluster Index Tree*(简记为 *CIT*).

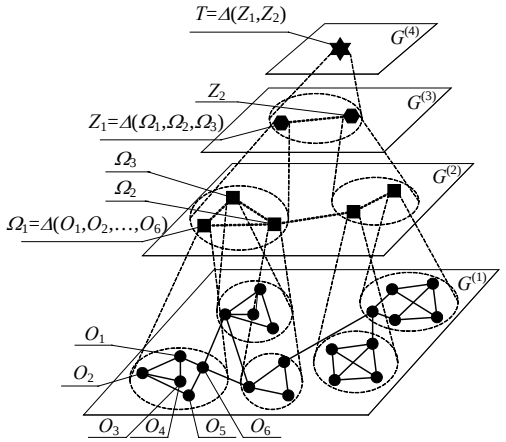

Fig.4 An example of index structure on ML^{2}O

图 4 一个 ML^{2}O 上索引结构的实例

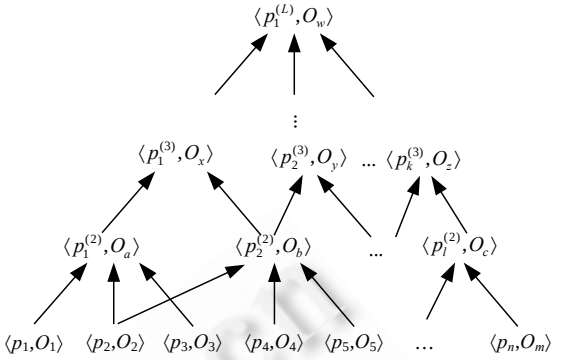

Fig.5 Cluster index tree (or *CIT*)

图 5 簇索引树(*CIT*)

显然,图 4 和图 5 给出的是一种全局索引结构.要建立如图 5 所示的索引树,需要已知:(1) 所有节点的索引信息$\langle p_i,O_j\rangle$;(2) 所有节点间的连接信息.这样的要求并不适合分布式 P2P 环境,P2P 网络中的任意节点 p 根据直接连接(即一阶邻居)只能获取索引信息$\{\langle q,O(q)\rangle|q\in N(p)\}$和连接信息$\{(p,q)|q\in N(p)\}$.在部分索引信息和部分拓扑信息基础上,只能建立部分 *CIT*.更确切地说,是建立起 *CIT* 某个部分的近似.算法 1 给出了在收集了部分节点的信息后建立部分 *CIT* 的算法,为方便起见,在节点 p 处收集了集合 N 中各节点的信息后建立的部分 *CIT* 记为 $p.CIT(N)$.

算法 1. $p.BuildCIT(N)$.

输入:$p,O(p),N\subseteq P,\{O(n_i)|n_i\in N\}$;

输出:$p.CIT(N)$.

(1) 初始化树 $T=\langle p,O(p)\rangle$

(2) for ($n_i\in N$) {$O_c=\Delta(O(p),O(n_i))$

(3) 从节点$\langle p,O(p)\rangle$沿父链找第 1 个满足 $O_c\subseteq O_d$ 的节点$\langle_,O_d\rangle$

(4) if (找到这样的节点,记其为$\langle_,O_d\rangle$) {

(5) if ($O_d=O_c$) {建立节点$\langle n_i,O(n_i)\rangle$并将其插入为$\langle_,O_d\rangle$¬的子节点}

(6) if ($O_d\neq O_c$) {找到$\langle_,O_d\rangle$子节点中所有满足 $O_e\subseteq O_c$ 的节点$\langle_,O_e\rangle$

(7) 建立节点$\langle_,O_c\rangle$和$\langle n_i,O(n_i)\rangle$,将$\langle n_i,O(n_i)\rangle$插入为$\langle_,O_c\rangle$的子节点

(8) 将所有$\langle_,O_e\rangle$插入为$\langle_,O_c\rangle$的子节点,断开$\langle_,O_d\rangle$和所有$\langle_,O_e\rangle$的连接

(9) 将节点$\langle_,O_c\rangle$插入为$\langle_,O_d\rangle$¬的子节点}

(10) } //对应语句(5)的 if

(11) if (没有找到这样的节点) {

(12) 建立节点$\langle n_i,O(n_i)\rangle$,并找到 T 的根节点 $r=\langle\_,O_r\rangle$

(13) if ($r=\langle p,O(p)\rangle$) {建立节点$\langle_,O_c\rangle$,将$\langle p,O(p)\rangle$和$\langle n_i,O(n_i)\rangle$插入为$\langle_,O_c\rangle$的子节点}

(14) else {建立节点$\langle_,O_c\rangle$,将$\langle_,O_r\rangle$和$\langle n_i,O(n_i)\rangle$插入为$\langle_,O_c\rangle$的子节点}

(15) } //对应语句(12)的 if

(16) } //对应语句(2)的 for

以图 3 的 ACM Topic 为例,设节点 p(其存储信息的 topic 为 Network topology)的邻居节点及其存储信息的 topic 为$\langle a$,Network topology$\rangle$,$\langle b$,Wireless communication$\rangle$,$\langle c$,Network Architecture and Design$\rangle$,$\langle d$,Distributed

databases⟩,⟨*e*,Linear programming⟩,⟨*f*,Virtual memory⟩,则按算法 1 建立的 $p.CIT(N(p))$如图 6 所示.显然,图 6 只近似地刻画了图 5 的一部分,而且图 6 中的内部节点都形如⟨_,*O*⟩.这是由于图 5 中内部节点⟨$p_j^{(i)}$,*O*⟩中的 $p_j^{(i)}$ 是为了形象说明 ML^{2}O 的结构特点,在进行 P2P 搜索时无实际意义,所以无需在 $p.CIT(N(p))$中维护这样的信息.

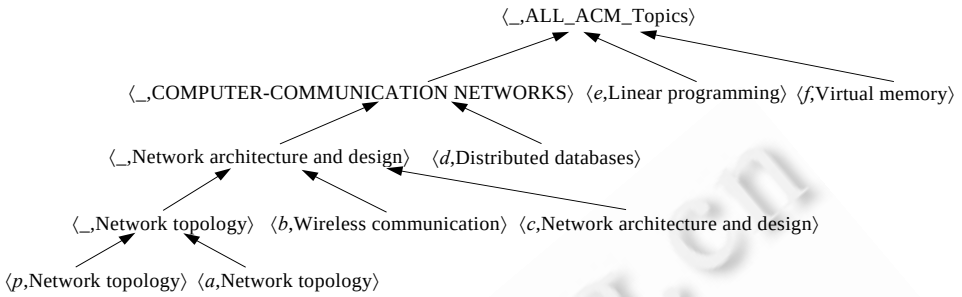

Fig.6 An example of $p.CIT(N(p))$

图 6 一个 $p.CIT(N(p))$实例

4 多级局部覆盖网上的搜索

4.1 定向渗透路由

基于 DHT 的结构化 P2P 搜索之所以能够获得良好的可扩展性,其根本原因是路由的每一步都有确定的转发方向,即知道该选择哪个节点转发才能更靠近目标节点.相比而言,无结构 P2P 网络上的路由是一种盲目的转发,路由的下一跳很有可能比当前节点更加远离目标节点.因此,要实现可扩展的无结构 P2P 搜索,必须在路由过程中找到更接近目标的下一跳节点,这是本文需要解决的问题.解决该问题时会遇到一个有趣的事实:结构化 P2P 网络的路由目标是很容易确定的(根据搜索请求的 ID 就能确定提供应答的目标节点的 ID),而无结构 P2P 网络却存在着本质的不同,路由的目标节点很难预先确定,通常在搜索完成后才确定.

“如何在未知搜索目标情况下找到更接近目标的下一跳节点”是本文在研究无结构 P2P 路由时要解决的核心问题.为此,本文提出了如图 7 所示的定向渗透路由机制(directed pervasion routing,简称 DPR).该机制有两个部分组成:渗透过程和定向过程.DPR 中的渗透部分可以采用 flooding 来得到局部节点的完整视图,也可以用 random walk 采样较大范围内节点的特征信息.根据这些信息,完成渗透后的节点可以计算出 DPR 中定向移动的下一跳节点.图 7 中的 *R-A-B-C* 就是 DPR 中的定向部分走过的路由,其余节点是用 DPR 中的渗透部分泛洪路由到达的.以采用局部泛洪来实现渗透的 DPR 为例,此时,搜索造成的通信负载为 $H\times K^{TTL}$,其中,*H* 是定向路由的跳数,*K* 是节点平均度,*TTL* 是 DPR 渗透部分的泛洪深度.由于 *K* 和 *TTL* 通常都是常数,所以 DPR 造成的通信负载是 $O(H)$,这表明依靠 DPR 实现可扩展的无结构 P2P 搜索的关键在于“在收集到的局部信息基础上,如何找到合适的路由下一跳”,这也是本文研究的基本诱因.

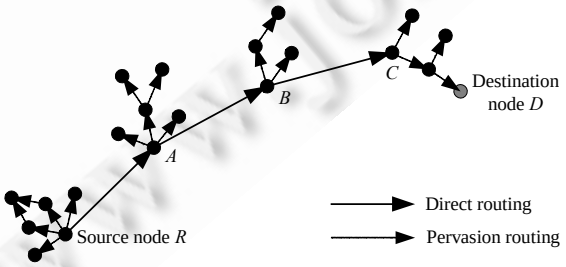

Fig.7 Directed pervasion routing (or DPR)

图 7 定向渗透路由(DPR)

4.2 多级局部覆盖网上的启发式搜索

在结合图 1 的多级局部拓扑 ML^{2}O、图 6 的簇索引树 *CIT* 和图 7 的定向渗透路由 DPR 的 3 项技术后,本文提出一种可扩展的无结构 P2P 搜索算法:Local Pervasion and Directed Search,简称为 LPDS.算法 LPDS 的输入是一个语义概念 O_r(就是用户输入的查询关键词,当用户输入多个关键词时,将多个概念分别作为 LPDS 的输入进行搜索,根据用户要求的多个关键词的逻辑关系进行搜索应答集合的相应操作,如用户的查询是多个关键词的 AND,则将搜索结果进行交运算),LPDS(O_r)的细节见算法 2.

算法 2. LPDS.

输入:r,O_r //发起查询的节点 r 和查询请求 O_r;

输出:无.

1. 对于节点 r 和收到定向搜索(类型为 *Directed Search*,简称 *DS*)数据包的节点

(1) if 当前节点是 r,则 C^B=*; else $C^B=pac_D.TOPC$ //C^B 定义了簇收缩的上界,即上面的 T'

(2) if ((*res*=*SearchInCluster*(RI,O_r))!=∅) {产生应答数据包给节点 r 发回 *res*}

(3) else {*next*=*SemanticIntegrate*(RI,O_r,& C^B);

(4) 向节点 *next* 发送数据包 pac_D=⟨*TYPE*=*DS*||*REQ*=O_r||*TOPC*=C^B⟩}

2. 函数 *SearchInCluster*(RI,O_r)

(1) *res*=∅; *RI*=∅; //*RI* 是一个三元组集合,渗透路由用 *RI* 来收集信息:节点,语义概念和计数值

(2) 广播数据包 pac_P=⟨*TYPE*=*PS*||*TTL*=1⟩ //*PS* 是 *Pervasion Search* 的简称

(3) 等待 pac_P 的应答,收到 pac_P 的节点 w 会返回信息{(p,*O*(p)),(q,*O*(q)),...} ($p,q,...$为 w 的邻居)

(4) 对每个收到的(p,*O*(p)),如果 *RI* 中没有 p 对应项,*RI* 加入新项(p,*O*(p),1);如果有,则将对应项计数加 1

(5) 将 *RI* 按计数分量从大到小排序,选取计数值明显大(超过阈值 Ψ_1)的那部分节点加入 *res*

(6) for ($z\in res$) if ($O_r\not\subset O(z)$) *res*=*res*/{z}; return *res*

3. 函数 *SemanticIntegrate*(RI,O_r,& C^B)

(1) *ttl*=0; *T*=∅; *RI*=∅; //*RI* 意义同上

(2) 向邻居节点广播数据包 pac_P=⟨*TYPE*=*PS*||*TTL*=*ttl*⟩ //*TTL*=0 表示直接收集节点 p 的 $N(p)$信息

(3) 根据收到的信息建立 *RI*,将 *RI* 按计数分量从大到小排序后的节点信息形成集合 *View*

(4) *T*=*BuildCIT*(*View*)

(5) if ($O(T.root)\supseteq O_r$) {

(6) 在 T 的所有节点中找到满足下式的节点 T' //$(\forall T)(O(T)\subseteq$*)

$$(C^B\supseteq O(T')\supseteq O_r)\wedge((\forall T'')(C^B\supseteq O(T'')\supseteq O_r)\rightarrow(O(T')\subseteq O(T'')))$$

(7) if (T'是叶节点) {*next*=T' 中的节点信息,$C^B=O$(*next*)}

(8) else {*next*=以 T' 为根的子树中任何叶节点中的节点信息,$C^B=O$(*next*)}

(9) else {

(10) *ttl*=*ttl*+1

(11) if (*ttl*>阈值 Ψ_2) {搜索失败}

(12) else {goto (2)}

(13) }

根据发起查询的节点 r 和应答查询 O_r 的节点 D 之间的关系(此处假定应答 O_r 的节点 D 均在簇 T 中),所以确切地说是根据 r 和 T 之间的关系,可将算法 LPDS 分为 3 个主要模块:(1) 簇内搜索(*SearchInCluster*):当 r 在 T 中或定向路由上的某个节点在 T 中时进行簇内搜索;(2) 语义综合(*SemanticIntegrate*):当 r 没有发现簇 T 时,首先需从 r 出发进行自下而上的语义信息综合(即语义索引的综合),语义综合结束后可找到同时包含 r 和 T 的簇 T',且 T' 的粒度一定大于簇 T 的粒度;(3) 簇收缩(ClusterContraction):找到簇 T' 后开始在 T' 中定位 T,这是一个自上而下收缩过程,在收缩过程中会调用 *SemanticIntegrate* 找到满足 $T\subseteq T''\subset T'$ 的 T'',这样,经过多次簇收缩(实

际上就是搜索区域的收缩)后就能定位到簇 T.

4.3 搜索算法LPDS的可扩展性分析

评价 P2P 搜索算法可扩展性的两个主要指标是搜索的应答时间和搜索造成的通信负载.对于 P2P 搜索,搜索的应答时间体现为从请求节点到应答节点的转发跳数,常用平均搜索跳数(average search hops,简称 ASH)表征.搜索造成的通信负载就是在搜索过程中引起的和搜索有关的通信包数量,可用每次搜索产生的通信包数(packets per search,简称 PPS)来表征[1−24].

定理 4. 如果发起请求的节点 r 在目标簇 T 中($r\in T$),LPDS 是可扩展的:$ASH=O(1)$,$PPS=O(1)$.

证明:对于 LPDS,节点 r 和收到定向搜索数据包的节点首先会进行簇内搜索 *SearchInCluster*,簇内搜索的核心是在邻居节点的信息 *RI* 中寻找计数次数最多的那部分节点(算法 2 函数 *SearchInCluster* 中的第(2)行~第(5)行),这部分节点形成 *res* 集合.

$r\in T$ 可以形式化地描述为$|N(r)\cap T|/|N(r)|\geqslant\delta$,其中,$\delta$近似为节点 r 的簇系数.由于 ML^{2}O 的局部性,δ是一个接近于 1 的常数.对于从 $N(r)$那里获得的$|N(r)|$个渗透应答信息,有大于$\delta\times|N(r)|$个渗透应答来自于 T 中的节点,给出的是 $t\in T$ 的邻居信息 $N(t)$.同样地,由于 t 处的局部性,$N(t)$中的节点有$\delta\times|N(t)|$个节点来自于 T.因此,在节点 r 收到的 $M\approx K^2$(K 是节点平均度)个渗透应答节点信息中,有$\delta^2\times M$ 个节点信息是来自于 T 的.所以,每个 T 中的节点在 *RI* 中的计数值约为$\delta^2\times M/|T|\geqslant\delta^2\times K$.另一方面,由于$(1-\delta^2)\times M$ 较小,且 *RI* 中非 T 的其他节点能取到情况有很多,所以这些节点在 *RI* 中的计数值相比很小.因此,当设定算法 2 中的 $\Psi_1=\delta^2\times K$ 时就能找到 T.

此时,只需 1 次 $TTL=1$ 的渗透路由即可.所以,$ASH=1=O(1)$,$PPS=K=O(1)$. □

定理 5. 当 $r\notin T$ 时,LPDS 仍然是可扩展的:$ASH=O(\log n)$,$PPS=O(\log n)$.

证明:$r\notin T$ 时的 LPDS 包含两个阶段:自下而上的语义信息综合阶段和自上向下的簇定位阶段.

从算法 2 函数 *SemanticIntegrate* 中的第(6)行可以看出,自下而上语义综合阶段的基本目标是在 *CIT* 中找到同时包含 r 和 T 的节点.以图 6 为例,如果 $r=p$,O_r=Distributed databases,则 T 就是节点 d 所在的簇.此时,LPDS 中的语义综合会找到节点⟨_,COMPUTER-COMMUNICATION NETWORKS⟩.

SemanticIntegrate 的可扩展性取决于算法 2 中的 pac_P 数据包的渗透深度 ttl,显然,*SemanticIntegrate* 造成的 $ASH=ttl$,$PPS=K^{ttl}$,其中的 ttl 是能成功找到同时包含 r 和 T 的 *CIT* 节点所需的泛洪深度.因此,分析语义综合阶段的扩展性等价于分析满足该条件的 ttl 值.由于 ML^{2}O 和 LPDS 存在随机性,所以分析 ttl 的期望值 $E[ttl]$:

$$E[ttl]=\sum_{i=0}^{\infty} i\times Pr(ttl\text{等于}i\text{能找到同时包含}r\text{和}T\text{的CIT节点}).$$

事件 E="成功找到包含 r 和 T 的 *CIT* 节点"可以用图 8 来刻画.从图中可以看出,成功找到 T'的充要条件是,在节点 r 用渗透路由收集的 *RI* 中包含簇 T''中的节点信息.

所以,$Pr(ttl=0\text{ 时发生 }E)=Pr(N(r)\cap T''\neq\varnothing)$.为简化分析,平均情况下可认为,节点 r 到 T'' 中任意节点 $t''\in T''$ 的距离都有$\kappa(r,t'')=\kappa(r,d)$.由公式(9),$Pr(t''\in N(r))=1/(\hat{C}^{\kappa(r,d)-1})$,所以,

$$Pr(N(r)\cap T''\neq\varnothing)=1-\left(1-\frac{1}{\hat{C}^{\kappa(r,d)-1}}\right)^{|T''|}\approx 1-\mathrm{e}^{\frac{-|T''|}{\hat{C}^{\kappa(r,d)-1}}}.$$

由图 8 和图 4 可以很容易地看出,平均情况下簇 T''包含 $\hat{C}^{\kappa(r,d)-1}$ 个节点,所以 $Pr(ttl=0\text{ 时发生 }E)\approx 1-1/\mathrm{e}$.

$Pr(ttl=1\text{ 时发生 }E)=Pr((\cup_{s\in N(r)}N(s))\cap T''\neq\varnothing)$,对于每个 $s\in N(r)$中的 $N(s)$都有 $Pr(N(s)\cap T''\neq\varnothing)\approx 1-1/\mathrm{e}$.所以,$Pr(ttl=1\text{ 时发生 }E)=(1/\mathrm{e})\times((1-(1/\mathrm{e})^{\hat{C}})$.类似地,$Pr(ttl=i\text{ 时发生 }E)=(1/\mathrm{e})\times(1/\mathrm{e}^{\hat{C}})\times\ldots\times(1/\mathrm{e}^{\hat{C}^{(i-1)}})\times(1-(1/\mathrm{e})^{\hat{C}^i})$.所以,

$$E[ttl]\approx\sum_{i=0}^{\infty} i\times\left(\frac{1}{e^{\hat{C}}}\right)^{\frac{i^2}{2}}\times\left(1-\left(\frac{1}{e^{\hat{C}}}\right)^{i}\right)\leqslant\sum_{i=0}^{\infty} i\times\left(\frac{1}{e^{\hat{C}}}\right)^{\frac{i^2}{2}}\leqslant c\sum_{i=0}^{\infty} i\times\left(\frac{1}{e^{\hat{C}}}\right)^{i}=\frac{ce^{\hat{C}}}{(e^{\hat{C}}-1)^2}.$$

因此,ttl 的期望值 $E[ttl]$是一个常数,表明 *SemanticIntegrate* 的 $ASH=PPS=O(1)$.

自底向上的语义综合阶段找到了 r 和 T 的公共祖先节点 T',更确切地说是找到了一个节点 $d'\in T''$(图 8 中的节点 d'),使得 *BuildCIT* 函数在将⟨d',$O(d')$⟩加入到 *CIT* 中时,*CIT* 中出现了包含 r 和 T 的节点 T'(算法 2 函数 *SemanticIntegrate* 中的语句(5)).语义综合结束后就能得到节点 d',而给节点 d'发送的定向路由包会在节点 d'处

执行下一轮语义综合.从图 8 可以看出,在 d'处执行语义综合得到的“同时包含 d'和 T 的 *CIT* 节点”是 T''.显然,T''满足 $T\subseteq T''\subset T'$,这就完成了 LPDS 的簇收缩.基于上面的分析,在 d'处执行 *SemanticIntegrate* 的 *ASH* 和 *PPS* 也是常数.因此,$r\notin T$ 时 LPDS 的 $ASH=PPS=O(1)\times$簇收缩次数.

由于每次簇收缩都有 $T\subseteq T''\subset T'$,再有 $T'\subseteq P(G)$,所以簇收缩次数至多为 *CIT* 的高$=L=O(\log n)$.

综合上面的分析,$r\notin T$ 时,搜索算法 LPDS 的 $ASH=PPS=O(\log n)$. □

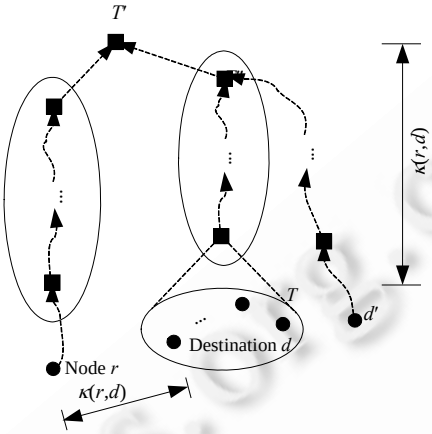

Fig.8 Situation of finding common ancestor of r and T in *CIT*

图 8 在 *CIT* 上发现 r 和 T 公共祖先节点的情况

定理 6. ML^{2}O 上的搜索算法 LPDS 是可扩展的.

证明:算法 2 表明,LPDS 算法的基本过程是,在每个定向路由中继节点处执行簇内搜索 *SearchInCluster* 和语义聚合 *SemanticIntegrate*.定理 4 表明,簇内搜索造成的 *ASH* 和 *PPS* 是常数;定理 5 表明,语义聚合造成的 *ASH* 和 *PPS* 也是常数.所以,每个定向路由转发节点造成的 *ASH* 和 *PPS* 仍是常数.定理 5 又表明,LPDS 中定向路由经过的节点个数为 $O(\log n)$.因此,ML^{2}O 上执行 LPDS 搜索造成的 $ASH=PPS=O(\log n)$,是可扩展的. □

定理 4~定理 6 分析的是信息完全一致的理想环境,在实际大规模动态 P2P 环境中,节点间维护的拓扑不一定严格符合定义 3 给出的 ML^{2}O 结构,节点上维护的索引信息也可能是错误的.这些因素都可能导致在寻找定向路由下一跳时没找到满足 $T\subseteq T''\subset T'$的簇 T'',使得实际工作环境中的 *ASH* 和 *PPS* 可能会显著大于 $O(\log n)$.但由于实际的 P2P 搜索请求具有局部性:局部存储信息的语义概念同时也是搜索请求的主要兴趣点[10−16], ML^{2}O 的拓扑结构、LPDS 的簇内搜索和局部语义聚合都能保证尽快找到局部范围内的应答目标.因此,本文提出的网络拓扑和搜索算法在实际 P2P 环境中也能够良好工作,本文用仿真实验验证了这一结果.

5 实验验证及结果

5.1 模拟实验平台

本文在覆盖网络模拟框架 PlanetSim[28]上实现了 ML^{2}O 覆盖网络,并在 ML^{2}O 基础上实现了定向渗透搜索算法 LPDS.为验证上述无结构 P2P 搜索算法的可扩展性,网络中的节点个数 n 从 100~10000 变化.另外,在验证搜索算法的性能时,本文以 ACM Topic ontology[27]中的概念作为 P2P 网络节点共享的内容,同时也以这些概念作为 P2P 查询的关键词.在网络初始化时,取出 ACM Topic ontology 全部叶子概念后,按均匀随机分布分发给网络中的每个节点,且把重复出现的概念全部删去(共分发了 906 个叶子概念),在不影响实验效果的同时增加结果的可比性.另一个影响实验结果的参数是公式(8)的簇规模 $\hat{C}$,对于本文的实验,将 ACM Topic ontology 中相同父概念下叶子概念的平均个数作为 $\hat{C}$ 的估计值,经统计,ACM Topic ontology 中的 1216 个叶子概念分别属于 258 个父概念,同父概念下叶子概念的平均个数为 4.7,所以将 $\hat{C}$ 的缺省值设为 4.

5.2 ML^{2}O的拓扑特征

图 9 给出了应用公式(10)建立节点连接时的网络拓扑结构,为了清晰地展现 ML^{2}O 的结构特点,该图给出的是 n=50,$\hat{C}$=3 时的网络拓扑.可以看出,基于公式(10)建立的网络拓扑在部分节点集上表现出了局部性及多级局部性.由于 ML^{2}O 的局部汇聚是一种统计特性,随着节点规模的增大,在越多的网络节点处会表现出局部特性.

图 10 表明,随着节点规模的增大,形成局部聚簇的节点数量迅速增加.当节点个数超过 2000 时,网络中几乎所有节点都局部汇聚,从侧面表明,用定理 2 的概率连接方法可以构建形成统计局部性的无结构 P2P 网络.图 11 从数量上分析了统计局部性覆盖网络 ML^{2}O 的两个重要的网络参数:节点度和节点间的距离.图 11 给出的是节点度和节点间距离随节点数量的变化,从数量结果可以看出,用本文的算法形成的覆盖网络,其节点度和节点间距离都随网络规模对数增长,且在网络中的各个地方表现出的特征基本相同,验证了定理 1 和定理 3 的正确性.

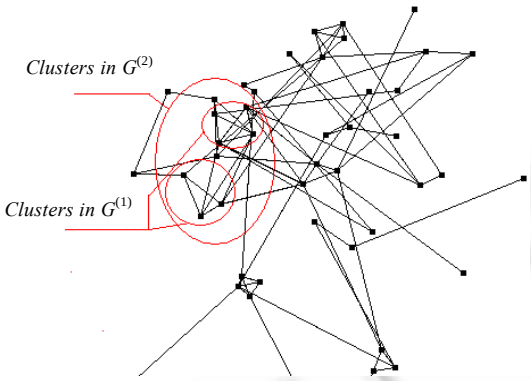

Fig.9 Network topology build by formula (9)

图 9 应用公式(9)建立的网络拓扑

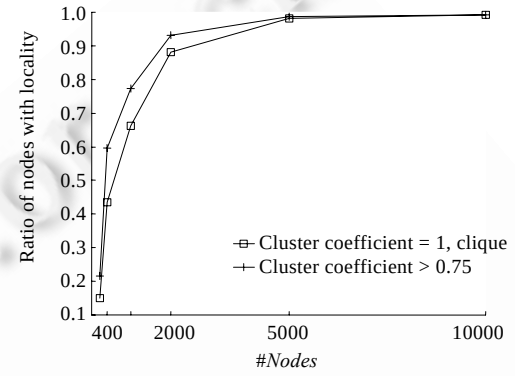

Fig.10 Ratio of node with locality and clustering

图 10 出现局部汇聚的节点比率

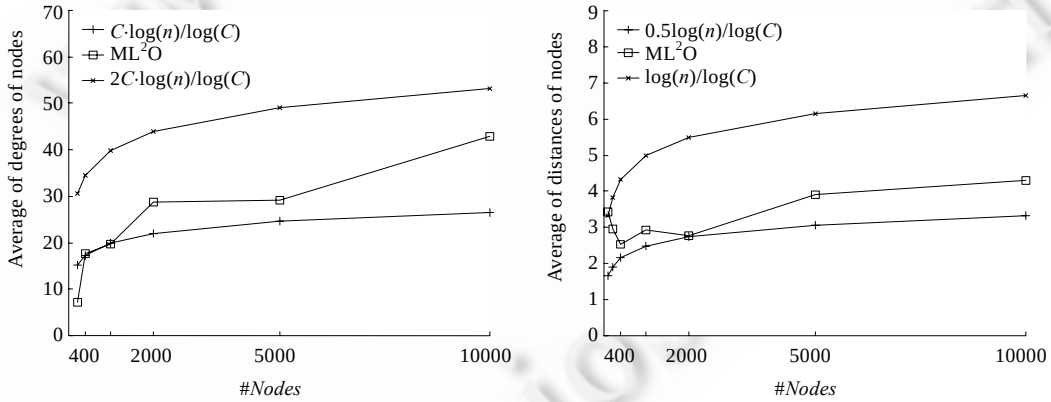

Fig.11 Degree and distance of nodes with different network scales

图 11 不同网络规模下节点度和节点间距离

5.3 LPDS搜索效率的可扩展性

用平均搜索跳数(即 *ASH*)和网络规模 n 的关系来对 LPDS 的可扩展性进行分析.为对 LPDS 的搜索效率给出更为有效的说明,此处对算法 2 进行了简单改造:节点在执行 *SemanticIntegrate* 中建簇树 T 的过程时只收集一阶邻居的信息,即如果在没有找到算法 2 第(6)行中满足条件的 T',则在其邻居节点中随机选取一个节点作为 *next*.在简化搜索过程的同时,给出了 LPDS 搜索效率的下界.另外,由于此时的 LPDS 搜索引起的通信负载是定

向路由 *ASH* 和节点邻居数的乘积,由于图 11 已给出节点平均度的分析结果,所以只需对比 *ASH* 即可.

图 12 和图 13 分别为在静态环境(没有节点的加入和离开)和动态环境(节点不断加入和离开)下,LPDS 和 Chord[17]的对比.为使结果具有可比性,Chord 中的节点也分发 906 个 key,也是从网络上随机选取的一个节点,该节点从 906 个 key 中随机选取一个 key 发起查询.图 12 的结果表明,当网络形成 ML^{2}O 拓扑结构时,LPDS 可以达到和 Chord 基本一样的搜索效率.另外图 12 也表明,LPDS 的 *ASH* 随网络规模增大,越趋近于理论分析的结果 $\log_2(n)$.这是因为网络规模越大,ML^{2}O 拓扑的统计特性就越明显.图 13 对比了动态环境下 LPDS 和 Chord 的搜索效率,此处用网络拓扑一致程度(用 topology consistency degree 或者 *tcd* 表示)来表征环境的动态程度,*tcd* 指网络中多大比例的边满足网络拓扑限制条件,静态环境就是 *tcd*=1 的特例.可用如下方法产生 *tcd*=λ的网络:在网络中随机选取(1–λ)×e 条边,将这些边的一个端点替换为另一个随机选取的节点.图 13 对比了多种 *tcd* 环境下节点数量为 10000 时 LPDS 和 Chord 的搜索效率.结果表明,Chord 对网络拓扑的不一致表现得很敏感,其搜索效率随 *tcd* 的增大接近指数速度下降,而 LPDS 的 *ASH* 随 *tcd* 的变化很小.即使在 *tcd* 较大时,LPDS 的搜索效率仍能保持在一个很好的水平上,说明 LPDS 作为无结构 P2P 搜索较结构化 P2P 搜索而言具有很高的鲁棒性.

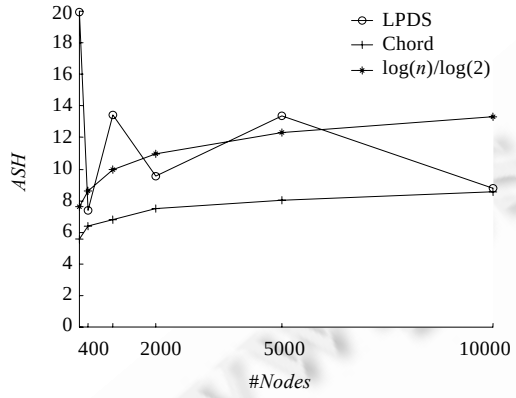

Fig.12 *ASHs* of LPDS vs. Chord under static environment

图 12 静态环境下 LPDS 和 Chord 的 *ASHs* 对比

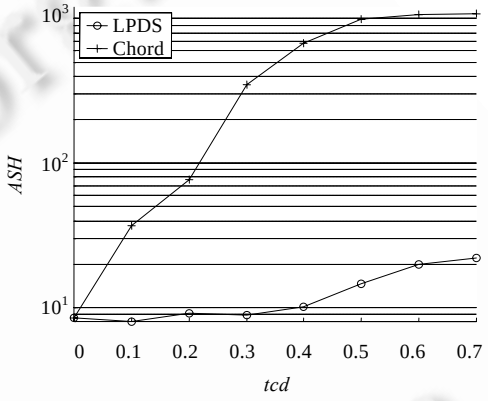

Fig.13 *ASHs* of LPDS vs. Chord under dynamic environment

图 13 动态环境下 LPDS 和 Chord 的 *ASHs* 对比

6 结束语

由于无结构 P2P 搜索具有很高的鲁棒性和环境适应性,所以许多实际的 P2P 系统都采用无结构 P2P 网络拓扑和无结构 P2P 搜索技术.但由于无结构 P2P 搜索的盲目性,搜索过程会产生大量的通信负载和很长的应答延迟,且这些效率指标通常都是关于网络规模的多项式,系统的可扩展性很差,限制了无结构 P2P 系统的应用范围.本文应用无结构 P2P 网络拓扑会随规模的增大而出现一定统计特征的性质,提出了一种被本文称为多级局部覆盖网络的无结构 P2P 网络——ML^{2}O,它借鉴了无结构 P2P 网络通常具有的局部特性,给出了能形成局部汇聚的节点间连接建立方法.同时,将这种连接进行适当的数学控制后,就能使产生的网络拓扑在从微观到宏观的多个粒度上呈现出局部性,这也是取名为多级局部覆盖网络的原因.本文对 ML^{2}O 的拓扑特性进行了理论分析,结果表明,ML^{2}O 的网络直径和节点平均度都是网络规模 *n* 的对数,为其上建立可扩展的无结构 P2P 搜索奠定了网络基础.由于 ML^{2}O 具有局部性,所以可以以局部为单位建立信息索引;而 ML^{2}O 在多个粒度上具有局部性,所以在局部索引的基础上为更大粒度的局部建立索引,就形成一棵索引树.在这棵索引树的分布式实现基础上,本文提出一种在局部渗透收集信息后建立部分索引树的基础上找到更接近搜索目标下一跳的搜索机制.理论分析表明,该搜索算法在搜索跳数和通信负载两个关键搜索评价指标上的数学期望都为 $O(\log n)$.另外,大量的模拟实验结果验证了理论分析结果,表明 ML^{2}O 上的搜索在动态环境下仍具有明显优于结构化 P2P 搜索的性能.

References:

[1] Meshkova E, Riihijärvi J, Petrova M, Mähönen P. A survey on resource discovery mechanisms, peer-to-peer and service discovery frameworks. Computer Networks, 2008,52(11):2097−2128. [doi: /10.1016/j.comnet.2008.03.006]

[2] Risson J, Moors T. Survey of research towards robust peer-to-peer networks: Search methods. Computer Networks, 2006,50(17): 3485−3521. [doi: /10.1016/j.comnet.2006.02.001]

[3] Bisnik N, Abouzeid AA. Optimizing random walk search algorithms in P2P networks, Computer Networks, 2007,51(6): 1499−1514. [doi: /10.1016/j.comnet.2006.08.004]

[4] Lü Q, Cao P, Cohen E, Li K, Shenker S. Search and replication in unstructured peer-to-peer networks. In: Proc. of the Supercomputing. 2002. [doi: /10.1145/514191.514206]

[5] Cohen E, Shenker S. Replication strategies in unstructured peer to peer networks. SIGCOMM Computer Communication Review, 2002,32(4):177−190. [doi: /10.1145/964725.633043]

[6] Terpstra WW, Kangasharju J, Leng C, Buchman AP. BubbleStorm: Resilient, probabilistic and exhaustive peer-to-peer search. In: Proc. of the SIGCOMM. 2007.

[7] Chawathe Y, Ratnasamy S, Breslau L, Lanham N, Shenker S. Making gnutella-like P2P systems scalable. In: Proc. of the SIGCOMM. 2003. 407−418. [doi: /10.1145/863955.864000]

[8] Puttaswamy KPN, Sala A, Zhao BY. Searching for rare objects using index replication. In: Proc. of the 27th Annual Joint Conf. of the IEEE Computer and Communications Societies (INFOCOM). 2008. 1723−1731. [doi: /10.1109/INFOCOM.2008.234]

[9] Crespo A, Garcia-Molina H. Routing indices for peer-to-peer systems. In: Proc. of the 22nd Int'l Conf. on Distributed Computing Systems (ICDCS 2002). 2002. 23−32. [doi: /10.1109/ICDCS.2002.1022239]

[10] Sripanidkulchai K, Maggs B, Zhang H. Efficient content location using interest-based locality in peer-to-peer systems. In: Proc. of the IEEE INFOCOM. 2003. 2166−2176. [doi: /10.1109/INFCOM.2003.1209237]

[11] Cai HL, Wang J. Exploiting geographical and temporal locality to boost search efficiency in peer-to-peer systems. IEEE Trans. on Parallel and Distributed Systems, 2006,17(10):1189−1203. [doi: /10.1109/TPDS.2006.139]

[12] Rostami H, Habibi J, Livani E. Semantic partitioning of peer-to-peer search space. Computer Communications, 2009,32(4): 619−633. [doi: /10.1016/j.comcom.2008.11.020]

[13] Tang CQ, Xu ZC, Dwarkadas S. Peer-to-Peer information retrieval using self-organizing semantic overlay networks. In: Proc. of the ACM SIGCOMM. Karlsruhe, New York: ACM Press, 2003. 175−186. [doi: /10.1145/863955.863976]

[14] Jin H, Chen HH. SemreX: Efficient search in a semantic overlay for literature retrieval. In: Proc. of the Future Generation Computer Systems 24 (2008). Elsevier Press, 2008. 475−488. [doi: /10.1016/j.future.2007.07.008]

[15] Rhea S, Geels D, Roscoe T, Kubiatowicz J. Handling churn in a DHT. In: Proc. of the USENIX Annual Technical Conf. 2004 on USENIX Annual Technical Conf. 2004. 10−24.

[16] Deerwester MS, Dumais ST, Furnas GW, Landauer TK, Harshman R. Indexing by latent semantic analysis. Journal of American Society for Information Science, 1990,41(6):391−407.

[17] Cohen E, Fiat A, Kaplan H. Associative search in peer-to-peer networks: Harnessing latent semantics. In: Proc. of the IEEE INFOCOM. 2003. 1261−1271.

[18] Stoica I, Morris R, Liben-Nowell D, Karger DR, Kaashoek MF, Dabek F, Balakrishnan H. Chord: A scalable peer-to-peer lookup protocol for internet applications. IEEE/ACM Trans. on Networking, 2003,11(1):17−32. [doi: /10.1109/TNET.2002.808407]

[19] Zhao BY, Huang L, Stribling J, Rhea SC, Joseph AD, Kubiatowicz JD. Tapestry: A resilient global-scale overlay for service deployment. IEEE Journal on Selected Areas in Communications, 2004,22(1):41−53. [doi: /10.1109/JSAC.2003.818784]

[20] Ratnasamy S, Francis P, Handley M, Karp R, Shenker S. A scalable content addressable network. In: Proc. of the ACM SIGCOMM. 2001. 161−172. [doi: /10.1145/383059.383072]

[21] Srivatsa M, Gedik B, Liu L. Large scaling unstructured peer-to-peer networks with heterogeneity-aware topology and routing. IEEE Trans. on Parallel and Distributed Systems, 2006,17(11):1277−1293. [doi: /10.1109/TPDS.2006.158]

[22] Li ZP,Huang JH,Tang H. A P2P computing based self-organizing network routing model. Journal of Software, 2005,16(5): 916−930 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/16/916.htm [doi: 10.1360/jos160916]

[23] Androutsellis-Theotokis S, Spinellis D. A survey of peer-to-peer content distribution technologies. ACM Computing Surveys, 2004, 36(4):335−371. [doi: /10.1145/1041680.1041681]

[24] Ioannidis S, Marbach P. Absence of evidence as evidence of absence: A simple mechanism for scalable P2P search. In: Proc. of the 28th Annual Joint Conf. of the IEEE Computer and Communications Societies (INFOCOM). 2009.

[25] Lü Q, Ratnasamy S, Shenker S. Can heterogeneity make gnutella scalable? In: Proc. of the 1st Int'l Workshop on Peer-to-Peer Systems (IPTPS). 2002. 94−103.

[26] Dean M, Connolly D, van Harmelen F, Hendler J, Horrocks I, McGuinness DL, Patel-Schneider PF, Stein LA. Web ontology language (OWL) reference version 1.0. http://www.w3.org/TR/2002/WD-owl-ref-20021112/

[27] The ACM topic. http://www.acm.org/about/class/ccs98-html

[28] PlanetSim. http://ants.etse.urv.es/planet/planetsim/

附中文参考文献:

[22] 李祖鹏,黄建华,唐辉.基于 P2P 计算模式的自组织网络路由模型.软件学报,2005,16(5):916−930. http://www.jos.org.cn/1000-9825/16/916.htm [doi: 10.1360/jos160916]

李治军(1977－),男,内蒙古伊盟人,博士,副教授,CCF 高级会员,主要研究领域为 P2P 系统,普适计算,操作系统.

李晓义(1981－),男,硕士,主要研究领域为 P2P 网络,激励机制.

姜守旭(1968－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为对等网络,传感器网络,普适计算.