

## PeerRank:一种无结构 P2P 资源发现策略\*

冯国富<sup>1,2+</sup>, 毛莺池<sup>1</sup>, 陆桑璐<sup>1</sup>, 陈道蓄<sup>1</sup>

<sup>1</sup>(计算机软件新技术国家重点实验室(南京大学),江苏 南京 210093)

<sup>2</sup>(南京审计学院 信息科学学院,江苏 南京 210029)

### PeerRank: A Strategy for Resource Discovery in Unstructured P2P Systems

FENG Guo-Fu<sup>1,2+</sup>, MAO Ying-Chi<sup>1</sup>, LU Sang-Lu<sup>1</sup> CHEN Dao-Xu<sup>1</sup>

<sup>1</sup>(State Key Laboratory for Novel Software Technology (Nanjing University), Nanjing 210093, China)

<sup>2</sup>(School of Information Science, Nanjing Audit University), Nanjing 210029, China)

+ Corresponding author: Phn: +86-25-83592339, Fax: +86-25-83300710, E-mail: fgfmail@dislab.nju.edu.cn, <http://cs.nju.edu.cn>

Feng GF, Mao YC, Lu SL, Chen DX. PeerRank: A strategy for resource discovery in unstructured P2P systems. *Journal of Software*, 2006,17(5):1098-1106. <http://www.jos.org.cn/1000-9825/17/1098.htm>

**Abstract:** One of the essential problems in P2P is the strategy for resource discovery. Related methods in unstructured P2P either depend on the flooding and its variations or utilize various indices, which results in too much overhead to forward messages or too expensive cost to maintain the indices. An adaptive, bandwidth-efficient and easily maintained search algorithm for unstructured P2P systems, PeerRank, is presented. The scheme utilizes the feedback from previous searches to probabilistically guide future ones. In addition, an effective caching and indexing mechanism is introduced, which remarkably enforces the search performance. The final simulation experiment shows that the strategy can remarkably improve the search efficiency with the small average path length, high success rates, very low bandwidth consumption, and the eminent adaptability to the change of hot resources.

**Key words:** P2P; resource discovery; forwarding strategy; adaptive caching; adaptive indexing

**摘要:** 资源发现是 P2P 应用所面临的最核心问题之一。相关的无结构 P2P 系统主要采用了查询消息泛洪和信息索引机制,这会造成严重的网络带宽负担以及巨大的索引维护开销。给出了一种无结构 P2P 环境下能够节约带宽、容易维护的自适应搜索策略 PeerRank。PeerRank 依据用户结点命中查询的历史信息赋予结点相应权值作为查询消息路由的依据,引导查询快速接近目标资源。自适应缓存机制和索引机制的引入使搜索性能大为加强。最后的实验表明,附带自适应缓存和索引的 PeerRank 以其高搜索成功率、多副本发现和很短的时间响应,能够显著地提高资源发现性能。

**关键词:** P2P;资源发现;转发策略;自适应缓存;自适应索引

中图法分类号: TP393 文献标识码: A

---

\* Supported by the National Natural Science Foundation of China under Grant No.60402027 (国家自然科学基金); the National High-Tech Research and Development Plan of China under Grant No.2004AA112090 (国家高技术研究发展计划(863)); the National Grand Fundamental Research 973 Program of China under Grant No.G2002CB312002 (国家重点基础研究发展规划(973))

Received 2005-02-28; Accepted 2005-08-25

P2P 提供了一种大规模异构环境下进行资源共享的有效途径,只要用户给出所需资源的属性描述,P2P 系统就能返回一组符合用户需求的资源列表.能否对用户请求作出成功而快速的响应直接决定着系统的可用性.而网络中大量点对点的通信使得 P2P 应用更加依赖于底层网络,每一次查询产生的消息数量直接影响着网络的通信负担,也最终决定着 P2P 方案的可扩展性以及可行性.因此,P2P 系统的一个最核心的问题就是以有效的手段快速地响应用户的资源请求,同时降低网络带宽消耗,保证系统可扩展性和容错性.

当前的 P2P 体系结构通常分为集中式、全分布方式和混合式.针对全分布体系结构的搜索算法都是通过结点之间转发、匹配查询消息来实现的,搜索算法的性能主要取决于查询消息所采用的路由策略.目前已有的路由策略主要分为基于信息路由和盲目消息路由两类.在基于信息的路由策略中,依据自身已有的邻居结点信息和资源信息,有区别地选择邻居结点转发查询,以提高发现目的资源的效率.在盲目路由方式中,采用广播或者随机选取部分结点进行转发的策略.盲目消息路由由算法简单、有效,但是网络开销太大.为此提出了许多算法,以减少广播算法的缺点.例如,在 Random walks 算法中设置  $k$  个 walker,walker 在网络中漫游遍历而不是扩散查询消息.

本文给出了一种基于信息的全分布体系结构 P2P 搜索算法 PeerRank. PeerRank 首先在应用层覆盖网络中部署  $k$  个 walker,以 walker 漫游遍历结点方式来实现资源发现.然后基于用户结点历史信息获取每个结点能够满足查询的概率权值,并根据概率权值进行查询消息转发,引导 walker 快速发现目标资源.此外,还引进了自适应动态缓存和索引机制,使搜索性能大为加强.最后的实验表明,拥有自适应缓存和索引机制的 PeerRank 以其高搜索成功率、多副本发现、很短的时间响应、低带宽消耗及其对热点的自适应性,能够显著提高资源发现性能.

本文第 1 节回顾相关工作.第 2 节给出消息转发策略以及自适应缓存和索引机制.第 3 节进行大量模拟实验和性能分析.第 4 节给出结论.

## 1 相关工作

根据结点之间的互连方式,P2P 可以分为结构化 P2P 和无结构 P2P 两类.结构化 P2P 搜索响应快,消息复杂度小,但需要以很高的代价维护既定拓扑,不能很好地适应结点高度动态的 P2P 环境.无结构 P2P 网络维护代价小,但会带来巨大的查询消息开销,造成网络流量负担.Gnutella<sup>[1]</sup>采用查询消息广播方式实现网络资源发现.基于广播的搜索算法简单、有效,但是网络开销太大.为此提出了许多算法,以减少广播算法的缺点.Modified-BFS<sup>[2]</sup>并非每步都转发给所有邻居结点,而是随机选择  $k$  个,以减弱消息扩散.在文献[3]中,查询者生成  $k$  个查询消息,每个消息在结点之间依次转发传递,实现搜索.这样的每一个消息被称为 walker,这种搜索算法也称为 Random Walks. Random Walks 能够显著减少搜索过程产生的消息数量,但是由于其盲目搜索随机转发的本质,搜索性能不能满足要求.不同于 Random Walks,PeerRank 采用基于信息的查询转发策略,根据历史信息提高查找效率.

此外,还有许多工作着重于提高查询消息转发的效率,即区分选择最有可能满足查询的结点进行转发.文献[4]根据 Power-law 拓扑,总是选择具有最大度的结点进行转发.文献[5]首先根据一定的标准判断两个查询的相似性,然后转发给最近满足相似查询的邻居结点.但总的说来,这些工作都只是为了快速发现目的文件,会导致巨大的网络开销.APS<sup>[6]</sup>(Adaptive Probabilistic Search)根据有效转发查询的次数赋予结点相应权值,作为转发查询消息的依据.但 APS 不能适应热点变化,不适合于资源访问频率不均的大规模 P2P 系统. PeerRank 以历史命中率作为路由依据,实验证明其更为合理、有效.

Freenet<sup>[7]</sup>通过大量的沿途文件复制和 LRU 文件管理算法实现相似标识文件聚集,形成基于标识符的路由方向感.但是,Freenet 大量的沿途复制文件会浪费大量的网络带宽,而且不能保证对非流行文件对象的查询结果.而且后面的实验也表明,Freenet 对深度搜索敏感,不适合宽度优先搜索,造成难以改进的响应延迟偏大.在 PeerRank 中,不需要大量的文件复制,同时结点对自己的存储空间拥有绝对的自治性.

## 2 基于概率权值的自适应查询消息转发策略

针对全分布体系结构的 P2P 搜索算法,最通用的方式就是将用户组织为一个应用层覆盖网络,通过在结点之间转发查询消息实现结点遍历查询.因此,消息转发策略,即以什么样的标准选取邻居结点转发查询消息,直接决定了 P2P 搜索性能.

### 2.1 搜索模型

假设一个具有  $n$  个结点和  $m$  个相异文件对象的简单模型系统,  $j$  结点拥有相异文件的集合是  $\rho_j$ , 系统所有相异文件对象的集合为  $O$ , 系统所有文件副本的集合为  $R$ . 用  $q_i$  代表系统对文件的访问频率,也就是在最近一段时间内,对文件  $i$  的访问次数占总访问次数的比例,则  $\sum q_i = 1$ , 其中  $i \in O$ . 我们按照  $q_i$  大小对文件进行排序编号,即  $q_1 \geq q_2 \geq q_3 \geq \dots \geq q_m$ .

平均情况下的一次查询,文件  $i$  被访问的概率为  $q_i$ , 那么,  $j$  结点所存储文件对象被访问的概率为  $\sum q_i (i \in \rho_j)$ . 因此,在  $j$  结点命中查询的概率为  $\text{Pr}_j, \text{Pr}_j = \sum q_i$ , 其中,  $i \in \rho_j$ .

因此,一个结点满足查询的概率不仅取决于所存储的相异文件对象的数量,还取决于所存取内容的流行程度.称获取结点 Pr 值的过程为 Peer Rank.

### 2.2 基于Peer Rank的查询消息转发策略

假设系统在最近一段时间共发起了  $Q$  次查询,对文件对象  $i$  的访问次数为  $Q_i$ , 则

$$Q = \sum Q_i, i=1,2,3,\dots,m,$$

$$Q_i = Q \times q_i, i=1,2,3,\dots,m.$$

假设每个结点收到查询消息的机会是等同的,  $M_j$  为平均情况下,  $Q$  次查询中  $j$  结点收到的消息数,每次查询平均遍历  $A$  个结点,则

$$M_j = QA/n.$$

设  $j$  结点命中的查询次数为  $t_j$ , 那么,

$$t_j = (\sum Q_i) A/n, i \in \rho_j,$$

$$t_j = (\sum q_i) QA/n = (\sum q_i) M_j = \text{Pr}_j M_j, i \in \rho_j.$$

所以,  $\text{Pr}_j = t_j/M_j$ , 即  $j$  结点在  $Q$  次查询中的命中率.

因此,  $j$  结点在平均情况下的 Pr 值等于该结点在过去一段时间内的命中率.下面用 Pr 值作为结点权值来衡量结点满足查询的概率,指引查询消息快速地到达目标结点、找到目标资源.我们称这种将结点 Pr 值作为消息转发依据的路由策略以及基于该策略的搜索算法为 PeerRank.

### 2.3 结点组织

当结点加入系统时,需要初始化路由表.最常用的方法是通过“半径递增”广播或者从一些知名的管理结点获取一个已知的在线结点,从而获取更多的邻居结点信息,逐步完成路由表构建,并将自己推给其他结点.

为了降低网络维护代价,结点离开系统时并不以实时方式通知与之相关的结点.当结点转发查询没有响应时,就认为该邻居结点离线,再次选取最优的邻居结点转发查询,同时修改路由表.

当结点收到查询请求时,更新 Pr 值,并携带 Pr 值应答,以便同步.为降低复杂度,可以时间片或者 Pr 值增量为单位同步更新 Pr 值.

### 2.4 转发策略的改进

#### 2.4.1 简单自适应缓存机制

上述路由策略的一个特点是保持文件位置不变,总是将消息转发到高权值结点.这样会导致部分热点成为性能瓶颈,同时并不能缩短普通查询结点和高权值结点间的距离.为此,一种可行的方案就是使热点文件在系统中具有更广泛的分布,加强对热点文件的命中效率,减小搜索距离,消除热点瓶颈.

这里引入简单自适应缓存机制.在一个结点命中查询之后,查询源将该文件复制到自己的缓存.这样,随着查询的进行,系统中将有较多的热点文件副本,能够快速命中大量针对热点文件的查询.

另一方面,这样不加选择地将命中的文件复制到本地,会使得同一文件的副本在某些结点周围出现聚集现象,占用非流行文件副本空间,造成“浪费”.对于 TTL(Time-to-Live)较小的宽度优先遍历算法尤其如此.因此,如果结点能够以很大的概率再次命中针对该文件的查询,就不需要缓存该文件;只有当距离命中查询的结点很远时才将文件在本地做缓存.假设在  $r$  跳之内命中的文件无须做缓存便能够再次命中对该文件的查询,则称  $r$  为辐射半径.

#### 2.4.2 自适应索引机制

通过简单自适应缓存机制可以知道,如果一个结点最近发起了针对一个文件的查询,那么该文件会以很高的可能性在它的缓存命中,或者通过转发查询消息再次命中.因此,可以为每个结点设置一个列表,记录最近所转发查询的查询源和文件标识,作为文件索引.因此,索引项能够反映系统的访问热点,加快响应速度,加强命中率,而且能够缓解部分瓶颈热点的负载.

缓存是有限的.每个结点采用 LRU(Least Recently Used)算法管理缓存空间和索引表项,缓存新的内容,同时淘汰最久不用的内容.这就带来索引和缓存内容同步的问题,即当索引项引导 walker 到一个结点时,如果该结点恰好已经把该文件从缓存里面删除,那么索引项会“误导”walker,造成索引失效.为此,为每个索引项加一个时间戳.只有时间戳在某个时间段范围之内时才启用索引项功能;否则,忽略索引项.假设每个索引项的平均淘汰周期为  $S$ ,如果索引项时间戳到当前时刻这段时间内淘汰索引项的次数小于该结点的存储空间,就启用该索引项;否则,忽略该索引项.

另一方面,walker 遍历的结点分布在查询者周围,如果在遍历结点都添加索引,会造成索引信息重复性地集中.当一个查询的 walker 遍历区域和集中索引区域相重叠时,多个 walker 会同时启用多个结点上的同一索引项,那么,就会将多个 walker 引向同一个结点,造成查询消息集中.这样,一旦在该结点命中失败,walker 搜索有效性立刻降低.我们采取概率的方式随机确定是否添加索引项.称这个概率  $P$  为索引项密度概率,它决定了添加的索引项数量和 walker 遍历的结点数量之间的比例.

这样,每一个结点需要维护 3 个列表:一个是邻居列表,用于构建整个逻辑覆盖网络;另一个是缓存列表,用于存储自己最近访问过的文件;最后一个索引列表,记录最近所转发查询的查询源和文件标识.下面是中间路由结点收到查询消息以后的处理过程伪码描述.

```

Procedure midPeer.Process ()
1. query=midPeer.GetQuery ()
2. resultList=midPeer.Retrieve(query)
3. if resultList!=NULL then
4.   midPeer.ModifyPeerRank ()
5.   return (resultList&peerInfo)
6. else
7.   TTL--
8.   if TTL>0 then
9.     indexItem=RetrieveIndex(query)
10.    if IsValid(indexItem)=true then
11.      relayPeer=indexItem.SourcePeer
12.    else
13.      relayPeer=SelectNextHop(query)
14.    end if
15.    midPeer.Forward (RelayPeer)
16.    InsertIndex()

```

```

17. end if
18. midPeer.ModifyPeerRank()
19. return (peerInfo)
20. end if

```

### 3 实验模拟

下面针对 PeerRank 做宽度优先遍历搜索和深度优先遍历搜索方面的大量实验,并与 Random Walks 搜索算法和 Freenet 搜索算法作性能比较.

#### 3.1 实验环境

应用层覆盖网络并不反映实际的物理网络.实验中,起初拓扑结构使用了基于 Waxman 算法的随机图,设定网络规模为 10 000 个结点,平均出度为 10.Freenet 的初始拓扑是一个环,每个结点将与之前后相邻的 4 个结点作为邻居.每个结点提供的文件数量采用比较均衡的几何分布模式.同一个文件副本数目的分布起初设定每个文件的副本为 90,在自适应缓存中,系统中副本的数目做动态调整.系统中资源被访问的频率按 Zipf-like( $\alpha=0.9$ ) 分布.发起查询的用户随机选取.更详细的信息可以参考文献[6-8].Walker 的最大数目设置为 12,略大于每个结点的平均邻居个数 10.每个索引项的平均淘汰周期  $S$  设置为 10 000 次查询,即平均每个结点被访问一次的时间段.设定 PeerRank 每个结点索引项数为 18,其总量是缓存空间的 2 倍左右.为了便于比较,设定 Freenet 每个结点的索引项数目为 28,即 PeerRank 的索引项数目与平均邻居个数之和.在与 Freenet 比较的实验中,TTL 取值为 200,以系统 70 次查询为一个时间片获取 DFS 搜索结果.

#### 3.2 实验结果

图 1 和图 2 显示了 PeerRank 两个参数对其性能的影响(其中, $walker=8, TTL=7$ ).图 1 是 PeerRank 搜索成功率随辐射半径的变化趋势.PeerRank 在辐射半径  $r$  取值为 2 时搜索成功率达到最高.这说明,2 跳之内的文件不必做缓存便能够以相当大的概率再次命中.图 2 是搜索成功率和索引密度概率之间的关系图.当  $P$  值为 0.05 时,搜索成功率达到最大值,比没有索引约提高 10%,比为所有经过的结点都添加索引时高 5 个百分点左右.下面的实验除非特别说明, $r$  取值为 3, $P$  取值为 0.25.

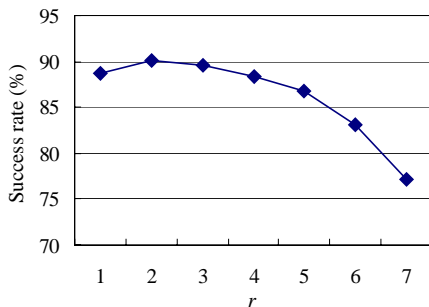


Fig.1 The success rate versus the parameter  $r$

图 1 辐射半径与搜索成功率之间的关系

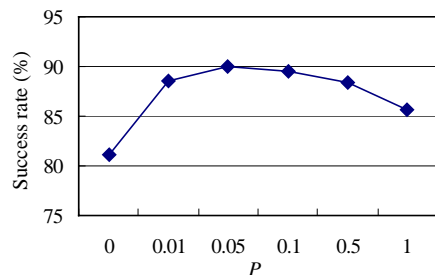


Fig.2 The success rate versus the parameter  $P$

图 2 搜索成功率与索引项密度概率之间的关系

如图 3 所示为 PeerRank 和 Random Walks 算法搜索成功率的比较.可以看出,与随机选取邻居结点进行查询消息转发的 Random Walks 算法相比,基于命中率进行查询消息转发的策略能够提高成功率 1 倍多,这是因为 PeerRank 能够根据以往满足查询的历史综合性地评估结点满足查询的概率,能够提高转发查询的有效性.PeerRank 和 Random Walks 在采取动态自适应缓存措施后搜索成功率都有大幅度的提高,尤其是 Random Walks 提高了 1 倍左右.这是因为动态的自适应复制能够使得热点文件得到充分扩散,能够提高针对热点文件大部分查询的搜索成功率.索引的引入使得 PeerRank 和 Random Walks 的搜索成功率也得到了大幅度的提高.各种算法的搜索成功率随着 walker 数目的增大而增长,但当大于 8 时出现 walker 冲突的机会增大,使得增长幅度

趋缓.

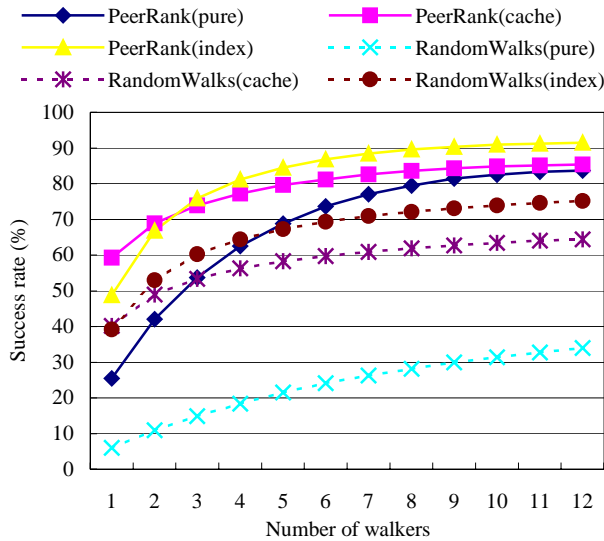


Fig.3 The success rate versus the number of walker

图 3 搜索成功率随配置 walker 个数的变化

图 4 是平均搜索路径长度,即搜索延迟的变化曲线.曲线说明,PeerRank 能够明显缩短平均响应时间.由于与上面相同的原因,自适应缓存机制的引入也能够相应提高响应速度.图 3 和图 4 说明,索引机制能够明显提高搜索成功率,walker 数目较多时更加明显,而对响应速度的改进却不明显.这一方面说明基于命中率的消息路由策略是有效的;另一方面说明索引机制能够比较有效地避免 walker 冲突,搜索成功率会有所提高,而由于索引项的非同步等因素使得路径“迂回”加长.

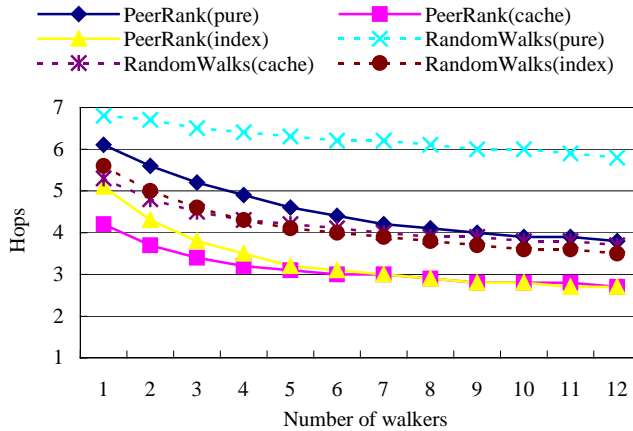


Fig.4 The average path length versus the number of walker

图 4 搜索响应跳数随 walker 个数的变化

一个好的搜索算法还应该发现更多文件副本,返回可供选择的列表.图 5 说明,具有自适应缓存和索引机制的 PeerRank 具有更高的 walker 利用率,即在部署相同 walker 个数的情况下,能够提供更多的可选资源.这是由于自适应缓存能够使得热点文件以更多的副本存在于系统,使得 walker 能够快速、有效地命中热点文件,具有更高的 walker 有效率.由于索引项和缓存的非同步以及索引项功能的“迂回”路径策略,索引机制虽然能够提高针对非流行文件的命中率,却不能更有效地提高大部分针对少量热点文件的访问.所以,引入索引机制的

PeerRank 的 walker 有效率反而降低,更严格的时间戳限制能够提高索引项内容与缓存内容的同步性,能够提高索引机制的 walker 有效率,但处于非同步状态的索引项就会增多,会将本来同步的索引项误认为非同步,浪费部分索引项空间,又在一定程度上削弱搜索成功率.对于 Random Walks,walker 之间的冲突程度和单个文件在系统中的副本数目对 walker 有效率的影响最大.

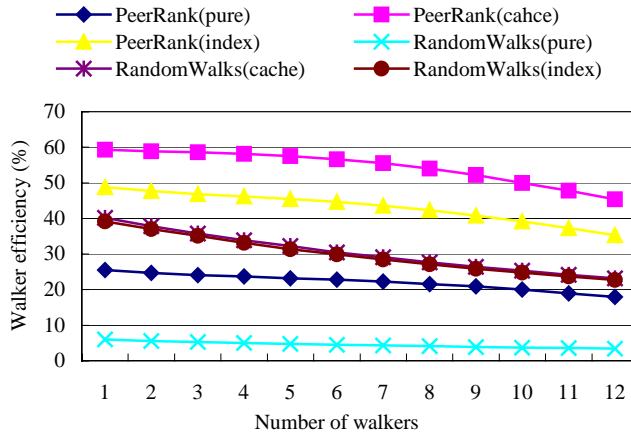


Fig.5 The efficiency per walker versus the number of walker

图 5 单 walker 效率与配置 walker 个数的关系

以上说明:基于结点 Pr 值来提高 P2P 搜索性能的效果是明显的;自适应缓存和索引机制对提高搜索成功率、提高响应速度和 walker 利用率也都是有效的.因此,查询消息将分布于查询者周围很小的半径范围,从而网络开销也将会得到改善.

图 6 和图 7 分别是 PeerRank(引入了自适应缓存和索引机制)和 Freenet 深度优先遍历搜索成功率和响应路径长度随时间的演化曲线.可以看出,PeerRank 的搜索成功率要高于 Freenet,基本维持在 90%以上.由于 Freenet 大量的沿途复制文件,使得热点文件副本在系统中迅速扩散,所以具有更快的收敛速度.又由于 Freenet 没有相应机制保证非流行文件在系统中的副本数量,大量热点文件副本“抢占”了非流行文件的存储空间,使得非流行文件在系统中副本数量逐渐减少,以至消失.因此,搜索成功率呈缓慢下降趋势.而 PeerRank 以索引优先,缓存其次的顺序处理查询,非流行文件在系统中消失的现象并不明显.统计发现,在 70 万次查询之后,文件的丢失率不足 5%,这对于非流行文件的逐渐更新过程来说是可以接受的.另一方面,Freenet 热点文件副本的大量复制使得大部分针对热点文件的查询拥有更小的搜索路径长度,最终的平均搜索路径长度和 PeerRank 相当.

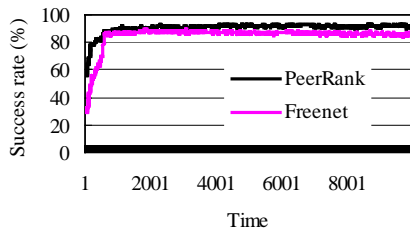


Fig.6 Time evolution of success rate

图 6 搜索成功率随时间的演化曲线

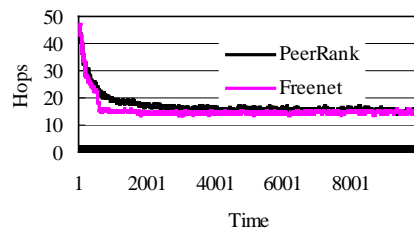


Fig.7 Time evolution of the average path length

图 7 平均搜索路径长度随时间的演化曲线

图 8 和图 9 分别是 PeerRank 和 Freenet 搜索成功率和平均搜索路径长度随时间变化的趋势,考察了 PeerRank 和 Freenet 的容错性能.

当系统基本达到收敛状态以后,每隔 10 万次查询随机抽取 20% 的结点,设定为离线状态,始终保持 80% 的用户在线.PeerRank 和 Freenet 在整体上都保持了良好的容错性.结点离线使得部分索引项失效,搜索成功率都下

降了 3% 左右,平均搜索路径增长了 1 跳左右.随着查询的进行,逐渐将包含离线结点的索引项淘汰,搜索性能逐步得到加强恢复,体现出了良好的容错性.

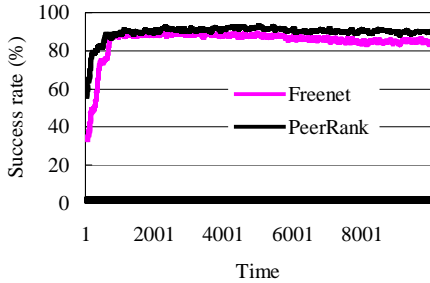


Fig.8 Time evolution of the success rate under the dynamic environment

图 8 容错环境下成功率随时间的演化曲线

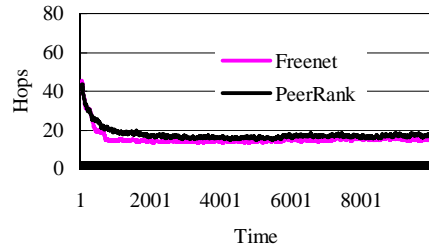


Fig.9 Time evolution of the average path length under the dynamic environment

图 9 容错环境下平均搜索路径长度随时间的演化曲线

图 10 和图 11 分别展示了 PeerRank 和 Freenet 对热点变化的自适应性.在 PeerRank 和 Freenet 基本达到收敛状态以后,我们每隔 10 万次查询随机选取 20% 的文件,对它们的流行度,即访问频率进行随机互换.可以看出,PeerRank 对热点变化的自适应性要远远好于 Freenet.Freenet 在热点发生改变以后性能降低明显,搜索成功率下降了近 30%,搜索路径长度增加了近 40 跳.这是因为 Freenet 的大量复制机制使得热点文件在系统中广泛分布,而非流行文件在系统中的副本数量很少以至于没有.当对原先不流行而现在流行的文件发起查询时,搜索往往失败而无法进行复制,使得性能下降而且不能很快恢复收敛状态.而 PeerRank 在热点发生变化后,搜索成功率略有下降,平均路径长度略有增长,而且能够很快恢复到收敛状态.这是因为 PeerRank 有两套路由机制,而且索引机制受到流行程度的影响较小.当缓存的作用削弱时,索引机制能够弥补自适应缓存的缺陷.而且 PeerRank 基于命中率的路由机制会随着最近热点的变化做动态的自适应的调整,所以在热点变化时维持了较高的性能.同时,PeerRank 的单点复制而不是沿途大量复制,相对减轻了热点变换带来的搜索性能恶化的程度.

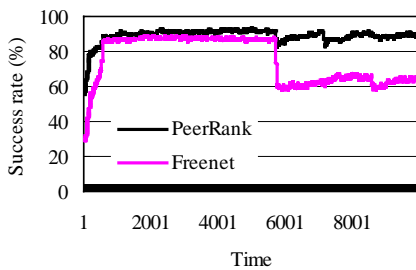


Fig.10 Time evolution of the success rate with the fluctuating hot-spots

图 10 热点动态变化情况下搜索成功率随时间的演化曲线

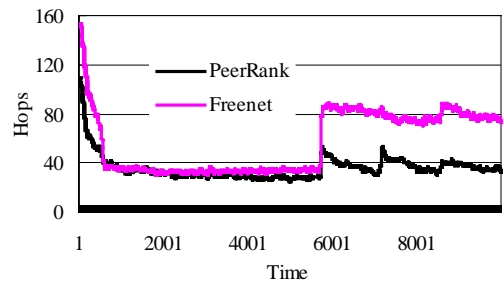


Fig.11 Time evolution of the average path length with the fluctuating hot-spots

图 11 热点动态变化情况下响应时间随时间的演化曲线

## 4 结 论

本文给出了一种无结构 P2P 环境下基于信息的资源发现策略,PeerRank. PeerRank 依据结点命中查询的历史信息动态地赋予结点相应权值,并基于权值进行查询消息路由,有效地引导查询接近目标资源.此外还引入了自适应缓存和索引机制,有效地加强了 PeerRank 的搜索性能.

实验表明,PeerRank 路由策略能够显著提高搜索性能,自适应缓存和索引机制能够加强 PeerRank 和



Random Walks 的性能,证明机制是有效的.与 Freenet 的实验比较也说明 PeerRank 具有很好的容错性和自适应性.

### References:

- [1] Ritter J. Why Gnutella can't scale. No, Really. 2005. <http://www.darkridge.com/~jpr5/doc/gnutella.html>
- [2] Kalogeraki V, Gunopulos D, Zeinalipour-Yazti D. A local search mechanism for Peer-to-Peer networks. In: Proc. of the 11th Int'l Conf. on Information and Knowledge Management (CIKM-02). New York: ACM Press, 2002. 300-307.
- [3] Lv Q, Cao P, Cohen E, Li K, Shenker S. Search and replication in unstructured peer-to-peer networks. In: Proc. of the 16th ACM Int'l Conf. on Supercomputing (ICS'02). New York: ACM Press, 2002.
- [4] Adamic LA, Lukose RM, Puniyani AR, Huberman BA. Search in power-law networks. Physical Review E., 2001,64(046135).
- [5] Ren Y, Sha C, Qian W, Zhou A, Ooi BC, Tan K-L. Explore the small world phenomena in pure P2P information sharing systems. In: Proc. of 3rd Int'l Symp. on Cluster Computing and the Grid (CCGrid). IEEE Computer Society, 2003. 232-239
- [6] Tsumakos D, Roussopoulos N. Adaptive probabilistic search (APS) for peer-to-peer networks. Technical Report, CS-TR-4451, University of Maryland, 2003.
- [7] Clark I, Sandberg O, Wiley B, Hong T. Freenet: A distributed anonymous information storage and retrieval system. In: Proc. of the Workshop on Design Issues in Anonymity and Unobservability. Heidelberg: Springer-Verlag, 2000. 311-320.
- [8] Iamnitchi A, Foster I. On fully decentralized resource discovery in grid environments. In: Proc. of the Int'l Workshop on Grid Computing. Springer Verlag Press, Germany, 2001.



冯国富(1977 - ),男,山东沂水人,博士生,主要研究领域为分布式计算,协议测试.



陆桑璐(1970 - ),女,博士,教授,主要研究领域为分布式并行计算,高性能计算.



毛莺池(1976 - ),女,博士生,讲师,主要研究领域为传感器网络,分布与并行计算.



陈道蔷(1947 - ),男,教授,博士生导师,CCF高级会员,主要研究领域为分布式计算.