

基于资源位置与节点反馈的 P2P 搜索算法^{*}

孙华志^{1,2} 侯洁²

(北京科技大学信息工程学院 北京 100083)¹ (天津师范大学计算机与信息工程学院 天津 300074)²

摘要 为了提高查询效率,本文提出了一种利用位置对节点分群,通过历史查询的搜索反馈结果动态选择转发与实际位置相邻节点的算法(FP 算法)。该算法通过计算邻居节点的兴趣相关度,定期调整邻居节点。算法分析和实验结果表明,与泛洪式算法相比本算法在搜索时间上改进约 10%~40%,同时很好地控制了总的消息数和重复访问节点的比例,提高了查询效率。

关键词 P2P, 搜索算法, 反馈, 兴趣相关度, 位置

P2P Network Search Algorithm Based on Position of Resources and Feedback of Nodes

SUN Hua-zhi^{1,2} HOU Jie²

(Information Engineering School, University of Science and Technology Beijing, Beijing 100083, China)¹

(College of Computer and Information Engineering, Tianjin Normal University, Tianjin 300074, China)²

Abstract In order to raise searching efficiency, this paper presents an algorithm that groups nodes based on their position and chooses to forward the searching demand to practically contiguous nodes dynamically according to their historical searching feedback results. Through calculating neighbor nodes' degree of interest correlation, the algorithm adjusts neighbor nodes regularly. The analysis of the algorithm and the results of experiments show that, compared to flooding algorithm, this algorithm improves performance in terms of searching time to 10%~40%, as well as controls the amount of messages forwarded and the proportion of repeating visit nodes, so it improves the searching efficiency.

Keywords P2P, Search algorithm, Feedback, Degree of interest correlation, Position

1 引言

无结构 P2P 网络的搜索算法^[1-7]可以分为两类,状态保存型和无状态类型。无状态类型的算法不保存任何关于网络状态和节点连接情况的信息;状态保存型算法为了改进随机搜索的效率会记录节点的状态信息。

文献[1]中提出基于距离的聚类方法,通过不断从候选节点中选择一个最近的节点,来减少 P2P 搜索的时间,但这种没有考虑节点兴趣的方法使得查询要经过较多节点,搜索结果返回缓慢。文献[2]中通过计算查询消息与转发节点共享信息的相似程度选择转发的节点,每次转发时都需要比较消息与共享信息的相似程度,造成搜索时间较长。文献[3]仅考虑了节点反馈结果,而忽略了节点地理位置,两个 ID 很近的节点在实际的地理位置可能较远,在 P2P 层的一跳到 IP 层有可能是多跳,这就存在请求延迟和下载延迟。

本文依据位置信息对节点分群,保存历史查询的搜索反馈结果,建立以兴趣相关度为基础的快捷兴趣表,计算邻居节点的预计搜索成功性,动态选择转发实际位置相邻节点,实验表明该算法可以显著减少带宽的占用和发现第一个文件的时间。

2 算法描述

2.1 位置分群

网络中传输延迟是普遍存在的,两个局域网 A, B 如图 1 所示。通常情况下子网内部节点之间的通信延迟很小,而 A 与 B 两子网节点之间的通信延时可能要达到几十或几百个

毫秒。如果子网 A 中某个节点发起搜索请求,该请求有可能首先转发至子网 B 或其它局域网内的节点,无形中加重了网络负载,增加了传输延迟时间,从而降低搜索效率。

分群就是把原本无序的节点按某一共同属性进行划分,基于位置信息的分群算法主要是通过计算普通节点和几个全局 Grid-mark 节点的传输延迟来进行分组,以保证查询转发过程中优先考虑地理位置距离较近的节点。

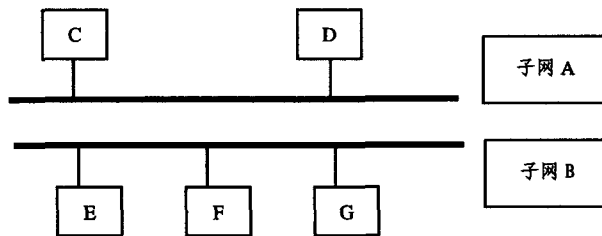


图 1 局域网示意图

本文提出的算法思想是在 P2P 网络中选定若干全局 Grid-mark 节点,这些节点具有性能稳定、处理能力强、网络通信质量好等特点,通过计算每个节点和全局 Grid-mark 节点间的传输延时来进行组群划分,延迟的大小可以代表距离的远近。

当有新的节点加入时,将临时变量 t_n 初始化为无穷大。新节点首先向网络中的所有 Grid-mark 节点发送消息,计算出自身和各 Grid-mark 节点的延迟,每一次计算出的延迟时间与 t_n 进行比较,若当前的延迟时间小于 t_n ,则记录当前 Grid-mark 的信息,将当前的延迟时间记入 t_n 。最终,将该节

^{*}天津市自然科学基金项目(编号:06YFJMJC00100)。

点加入当前记录 t_n 所对应的 Grid-mark 的群。考虑到某些特殊节点可能存在没有延迟或延迟很小的情况,选取一个阈值 ϵ ,如果计算出的延迟小于 ϵ ,则规定该节点可任意加入一个群组,所有节点排列顺序在系统初始化时确定。

对于有 n 个 Grid-mark 全局节点的 P2P 网络,将节点分为 n 个群组 G_1, G_2, \dots, G_n ,对于任意 $G_i (i=1, 2, \dots, n), G_i \cap G_j = \emptyset (i \neq j), G_1 \cup G_2 \cup \dots \cup G_n = U$ 。经过划分后,每个节点可以确定地属于某一个群,且同一个群内的节点都具有相近的位置信息。通过对群中 Grid-mark 的延时进行分类,可得到更为详细的节点位置信息。

2.2 节点存储的内容

网络中每个节点都保存关键字表、历史查询反馈表、当前查询信息表、历史查询的总个数和邻居节点路由表及本节点地理位置信息。

关键字表 CK: $CK[A] = \{ck_{ij}\}$ 表示节点 A 存储的文件的关键词, ck_{ij} 表示节点存储的第 i 个文件的第 j 个关键词, $i=1, \dots, m, j=1, \dots, n, m$ 为存储的文件个数, n 表示第 i 个文件的关键词个数。每个记录项有两个字段,即文件标识及其对应的关键词。

历史查询反馈表 CX: 该表保存查询命中的历史记录,存储着关键字及对应的命中节点信息。

快捷兴趣表 IST (Interest Shortcut Table), 保存与本节点有快捷链接的兴趣节点信息。该 IST 表由 interest, flag, num, address 字段组成,其中 interest 代表节点查询的兴趣, flag 是快捷连接的标志(取值为 0 或 1), num 代表查询成功的次数, address 代表快捷链接的节点位置。

当前查询信息表 $U_A = \{u_1, u_2, \dots, u_s\}$, U_A 表示节点 A 的当前查询信息, u_j 表示查询消息的第 j 个关键词, $j=1, \dots, s$ 。

另外还应保存着节点 A 历史查询的总个数 T_A , 邻居节点路由表,本地节点地理位置 P 信息。

2.3 兴趣相关度

根据文献[4,5]的研究成果,节点 B 存储的内容越多,并且节点 B 存储的内容与节点 A 存储的内容越相似,则 B 提供的查询命中率也越高,也即节点 B 对节点 A 越相关。因此提出兴趣相关度 $Y(A, B)$ 的计算公式:

$$Y(A, B) = S(A, B) / M * O \quad (1)$$

$$S(A, B) = \sum \left(\left(\sum_{i_1=1}^M \sum_{i_2=1}^N ck_{i_1 i_2} [A] \right) \cap \left(\sum_{j_1=1}^O \sum_{j_2=1}^N ck_{j_1 j_2} [B] \right) \right) \quad (2)$$

$S(A, B)$ 代表节点 A 与节点 B 存储文件的关键词相同个数, $i_1=1, \dots, M, i_2=1, \dots, N, j_1=1, \dots, O, j_2=1, \dots, N, M$ 代表节点 A 存储文件的个数, O 代表节点 B 存储文件的个数, N 代表每个文件关键词的个数, \cap 代表取交集,如果关键词相同结果为 1,不同结果为 0。

2.4 预计搜索成功性

实际上,历史查询结果对于新的查询是非常重要的^[4],大多数查询结果由较少具有高查询提供率的节点提供^[5]。因此,若认为节点 A 的第 b 个邻居对于节点 A 过去查询的平均提供率越高,且过去成功提供的查询与新查询越相似,则节点 A 在该节点上的预计搜索成功性越高,由于还要考虑节点的实际地理位置信息,则计算预计搜索成功性 $X(A, cg_b)$ 为:

$$X(A, cg_b) = T_p[b] * L_b / P_{a-b} \quad (3)$$

$$T_p[b] = cg_b / T_A \quad (4)$$

$T_p[b]$ 代表 A 的第 b 个邻居节点的平均查询提供率, L_b 代表 A 的第 b 个邻居节点的近期历史查询消息的关键词与新查询关键词的相同个数, P_{a-b} 为表示节点 A 到第 b 个邻居节点的实际位置距离的相关量, cg_b 代表 A 的第 b 个邻居节点成功提供结果的次数, T_A 代表节点 A 查询的总个数。

2.5 快捷兴趣表 IST

算法的关键是兴趣节点的寻找和快捷链接的建立。如当节点 A 发出的搜索请求,在节点 B 中命中,表明 A 与 B 存在有相似的兴趣,当 A 的后继查询多次在 B 中搜索成功,成功率达到某个阈值时,在 A 与 B 之间建立快捷链接(shortcut)。

2.5.1 快捷链接的发现

当一个节点 A 加入网络时, A 根据本地的共享资源建立本地信息表,该表根据本地共享资源的变化而进行更新。当 A 第一次发出查询请求时,它可采用某种泛洪搜索机制在网络中传播,同时返回一系列拥有该资源的节点表,这些节点都是快捷链接的候选者。只有当 A 成功地从 B 中下载资源时,对应的本地快捷兴趣表的 num 加 1,当 A 后续的查询继续在 B 搜索成功并下载完成 num 继续加 1,当 num 达到某个阈值时,本地兴趣表的 flag 置为 1,表示快捷链接建立。同时快捷链接还可通过节点的快捷链接建立,如节点 A 向 B 发出查询请求, B 如果存在快捷链接就通过自己的快捷链接进行查询,搜索成功后将信息返回给节点 A , A 成功下载后直接在本地兴趣表中追加记录,并直接建立快捷链接,并将 num 赋予规定的阈值。

当利用快捷链接搜索信息时,搜索失败时,则将 num 减 1,当 num 小于某个阈值时,就取消它们之间的快捷链接,并将对应表项 flag 置 0。

2.5.2 快捷链接的选择

当本地快捷兴趣表建立后,节点按照快捷链接的使用情况进行排队。如果某个快捷链接经常被使用,就将它放在缓存中,这样当节点进行搜索时首先调用缓存的快捷链接。其它按使用的情况在快捷兴趣表中排队,经常使用的排在上面。

同时也可以根据节点的命中率、带宽、时间延时等综合考虑,最常使用的快捷链接放在最上面。每个节点的快捷兴趣表可以自我调整,以适用 P2P 网络的变化。当一个快捷链接节点不存在,长时间不使用,或命中率很低,应调整它存放的位置及 num 的值,直到最后将其删除,加入其它新的快捷链接节点。

每个节点维持一定数目的快捷链接,当快捷链接的个数大于某个阈值,根据节点的调整策略进行选择,以维护节点设备的性能。

利用快捷兴趣表可以减少查询时间,对于查询消息属于该节点兴趣的查询效率明显较高。

2.5.3 快捷兴趣表的相关算法

• 建立快捷链接算法

①如果 A 是新加入节点,转②,否则转④;

②采用泛洪式,找到一些兴趣资源列表,修改本地兴趣表;

③判断本地兴趣表的某个节点的 num 是否大于某个阈值,若大于,则建立快捷链接,本次建立结束;否则对 num 进行调整转②;

④通过节点快捷链接的快捷链接建立新的快捷链接。

• 快捷链接的选择算法

①首先根据本地兴趣表的 num 进行排队,选择最上面的快捷链接;

②当按快捷链接查询时,按下载成功或失败对 num 进

行调整;

③ 当 num 小于某个阈值时,删除快捷链接。

2.6 搜索算法

如果 A 是新加入节点,位置分群;定期计算 A 与历史查询反馈表中节点的兴趣相关度,判断其是否大于某个阈值,若是建立快捷链,根据本地兴趣表的 num 进行排队,选择最上面的快捷链接;查找快捷兴趣表 IST,若找到算法结束,否则查询历史纪录反馈表,计算预计搜索成功性,从大到小选择 n 个转发的节点;搜索失败通过泛洪进行资源查询。下面给出的是查找包含关键字节点的算法描述,其中 A 是发起查询的节点,IST 为快捷兴趣表,Output 为查找出的节点。

```

1 Initialize: 初始化位置分群策略,确立全局 Grid-mark 节点
2 Begin//找到符合条件的节点
3 if A is new//A 是新加入节点
4   distribute(A,ε) //位置分群, ε 为系统设定的阈值
5 else
6   while( not the last record of the CX)
7     calculate Y(A,X) //计算兴趣相关度
8     if Y(A,X)> Threshold// Threshold 为系统设定阈值
9       insert(A,X) //插入快捷兴趣表
10    end if
11    while(not the last record of the IST)
12      if ture
13        output //查询结束,结果输出
14      else
15        for i in CX
16          calculate X(A,i) //依次计算节点 A 到第 i 个
17            result=order(X)//排序并按结果转发,查找
18              成功 result 置为 1
19          end for
20          if result is not true
21            flooding(); //泛洪式查找
22          end if
23        end while
24      end while
25    end if

```

3 实验及结果分析

为了验证本文提出的算法(FP 算法)的性能,实验采用 Waxman 提出的网络拓扑模型,所产生的网络节点随机分布于矩形网格上,模型中两个节点之间以某个概率决定是否将它们连接起来,而这个概率只与节点之间的距离有关并服从泊松分布。

通过利用 BRITE 仿真工具产生 waxman 随机拓扑模型并利用 Java 语言编写算法程序,分别测试节点数目在 200, 500,1000,1500 情况下算法的性能。实验环境为 Linux Red-Hat9.0, IntelP4 CPU 3.0GHz, 512MB 的内存,开发工具为 JCreator Pro 3.10,随机生成网络拓扑。

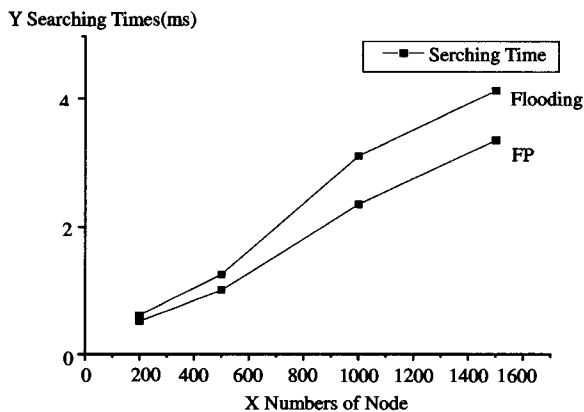


图 2 不考虑节点之间传输延迟泛洪式查找算法与 FP 算法的比较

仿真实验分为两组,第一组实验的目的在于不考虑实际物理位置的前提下,比较 FP 算法与泛洪式查找算法的执行

效率(如图 2),由于 FP 算法在节点转发的时候考虑节点之间兴趣相关度,并通过动态维护快捷兴趣表保存兴趣相关度较高节点的信息,因此减少了查询的转发次数,效率较高。

第二组实验的目的是在考虑节点之间传输延迟的前提下,比较 FP 算法与泛洪式查找算法的执行效率(如图 3),由于 FP 算法考虑了节点的实际地理位置并采用了分群的策略,确保节点在转发的过程中不仅考虑节点间兴趣的相关度,且优先将查询提交至实际物理距离较近的节点,从而减少了传输延迟,提高了查询效率。

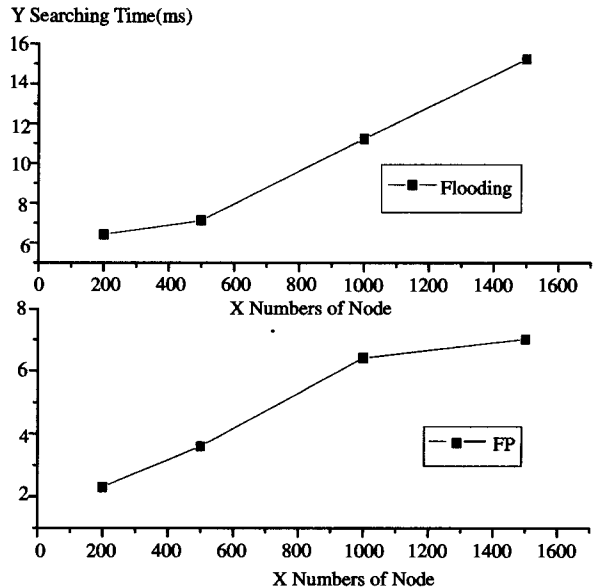


图 3 不考虑节点之间传输延迟泛洪式查找算法与 FP 算法的比较

图 2,图 3 的实验结果表明,FP 算法利用历史查询的搜索反馈结果,预计邻居节点的搜索成功性,动态选择转发的邻居节点,可以减少传送消息数量和发现第一个文件的时间;而泛洪式查找算法在选择转发的邻居节点时,没有考虑节点间实际物理距离以及兴趣的相关性,因此搜索性能较差。

结束语 本文提出的 P2P 搜索算法利用历史查询的搜索反馈结果,考虑节点物理位置,计算邻居节点的预计搜索成功性,动态选择转发的邻居节点以减少转发消息数量以及发现文件的时间;通过计算邻居节点的兴趣相关度,定期调整快捷兴趣表,提高了 P2P 网络的搜索稳定性。

参考文献

- [1] Zheng Wei, Zhang Sheng, Yi Quyang, et al. Node clustering-based on link delay in P2P networks [C]. ACM Symposium on Applied Computing, New York; ACM Press, 2005;744-749
- [2] Idreos S, Koubarakis M, Tryfonopoulos C. P2P- DIET: an extensible P2P service that unifies ad-hoc and continuous querying in super-peer networks [C]// Proc. of the ACM SIGMOD International Conference on Management of Data. New York; ACM Press, 2004; 933-934
- [3] 张晓玲,钟诚,李智,等. 基于反馈选择的 P2P 网络搜索算法[J]. 微电子学与计算机,2007,24(9):106-108
- [4] Sripanidkulchai K, Maggs B, Zhang Hui. Efficient content location using interest-based locality in peer-to-peer systems [J] // Proc. of the ACM SIGCOMM. New York; ACM Press, 2003;175-186
- [5] Guo Lei, Jiang Song, Xiao Li, et al. Fast and low-cost search schemes by exploiting localities in P2P networks [J]. Journal of Parallel and Distributed Computing, 2005, 65(6): 729-742
- [6] 杨振会,程楠. 非结构化 P2P 网络的资源搜索算法研究[J]. 现代计算机,2007,241:128-130
- [7] Koo S G M, Kannan K, Lee C S G. On neighbor-selection strategy in hybrid peer-to-peer networks [J]. Future Generation Computer Systems, 2006,22:732-741
- [8] Ganesan P, Sun Qixiang, Garcia-Molina H. Yappers: A Peer-to-Peer Lookup Service over Arbitrary Topology