第 34 卷 第 11 期 计 算 机 工 程 2008 年 6 月
Vol.34 No.11 Computer Engineering June 2008
·网络与通信· 文章编号：1000—3428(2008)11—0102—03 文献标识码：A 中图分类号：TP312

基于兴趣相关度的 P2P 网络搜索优化算法

吴 思，欧阳松
(中南大学信息科学与工程学院，长沙 410083)

摘 要：P2P 网络中的搜索性能是影响 P2P 网络发展的关键问题。该文研究非结构化分散型 P2P 网络中的搜索机制，提出 2 个改进算法。改进算法利用节点的共享情况和查询历史发掘节点的兴趣爱好，并赋予节点一定的自治性，使得非结构化分散型 P2P 网络能随着网络中查询数的增长而动态优化，提高查询效率。实验证明改进算法提高了查询检索的效率，在保证查全率的基础上，查询产生的消息减少了 75%。
关键词：P2P 网络；非结构化；搜索

Improved Search Algorithms Based on Interests Correlation in P2P Network

WU Si, OUYANG Song
(School of Information Science and Engineering, Central South University, Changsha 410083)

【Abstract】 Search performance is the key for the development of P2P network. This paper studies search mechanism in unstructured P2P network and introduces two algorithms which improve the efficiency of search. The algorithms make full use of sharing situation and the history of queries in peers, which dynamically improve the P2P network with the growth of queries. Experimental results show that the amount of messages created by queries is declined by 75%, which proves that the algorithms improve the efficiency of search.
【Key words】 Peer-to-Peer(P2P) network; unstructured; search

网络凭借着自身的优势获得了越来越广泛的应用，已成为研究的热点。P2P网络中的关键技术是文件搜索[1]。目前的P2P网络按照搜索机制可分为结构化和非结构化 2 大类。非结构化P2P网络模型按照拓扑结构又可以分为 3 类：集中模型(Napster[2])，分散模型(Gnutella[3])和混合模型(EDonkey)。本文描述了一种在非结构化的具有分散模型拓扑结构的P2P网络中查询检索优化方法，并建立原型系统测试了优化算法的性能，用实验证明该方法是有效的。

1 基于兴趣相关度的自组织的查询算法

非结构化分散型 P2P 网络中的搜索是一种广播式的搜索，如 Gnutella。网络中的节点收到查询以后不仅进行本地搜索，而且将查询发送到除查询来源节点以外的所有邻居节点。搜索范围由 *TTL*(Time To Live)决定：查询每转发一次，*TTL* 减 1，当 *TTL* 等于 0 时，查询便不再被转发。每个查询有唯一的编号，如果节点收到先前已收到的某一查询，说明查询路径已经产生了回路，节点就不再转发查询。广播式的查询比较简单，当 *TTL* 取值适当时，查询具有很高的查全率，能满足用户的查询需求。但是因为采用广播的方式转发查询，很容易造成查询洪流，引起网络拥塞，降低 P2P 网络的性能。对此，本文提出了一种改进的查询优化算法。该算法通过发掘 P2P 网络中节点的兴趣爱好，将兴趣相近的节点聚集起来，使得查询在有限的跳数里获得更高的查全率。同时，节点也被赋予了一定的自治力，可以通过探测周围的网络状态作出转发选择：广播查询还是选择性的转发查询。

在描述本文的算法前，先定义一个算法中用到的数据结构——兴趣域(InDomain)。

兴趣域由向量表示，表征节点的兴趣类别和对兴趣类别的感兴趣程度。一个兴趣域 $\boldsymbol{D}$ 表示为

$$\boldsymbol{D}=(d_1, d_2, d_3, \cdots, d_n)$$

其中，$d_k(k\in(1,n))$是非负实数，代表一个兴趣类(如计算机、天文、医学)，用于衡量节点对该兴趣类的感兴趣程度，d_k的取值可以有多种算法；兴趣域向量的长度n取决于兴趣类别的数量。

在本文的算法中，P2P网络中的每个节点都拥有一个自身的兴趣域和一个邻居节点表(存储邻居节点的IP地址和邻居节点的兴趣域)。算法中d_k的值为节点共享的每一类资源的数目，如节点p所在P2P网络中共有 4 个兴趣类别：计算机，天文，医学，文学；节点p共享了 13 个和计算机有关的资源，6 个和文学有关的资源，则节点p的兴趣域是(13, 0, 0, 6)。这样统计的前提是：节点共享的资源需要被划分到相应的兴趣类别中。如果资源是文档类型的，可以通过提取关键字实现：去掉文档中的虚词，提取有意义的词语，并计算每个词语出现的频率，频率最高的前m个词语被认定为关键字，拥有最多关键字的类别就被认为是该文档的类别。比如一个文档中的关键字是：P2P，查询，树，哈希，那么这篇文档就可以被划分到计算机类中。如果资源是非文档类型的，就需要用元数据来描述资源。可以通过提取元数据中“分类”一项的值来获得资源所属类别。

对于 2 个兴趣域分别为 $\boldsymbol{D}_1(d_{11},d_{12},d_{13},\cdots,d_{1k})$ 和 $\boldsymbol{D}_2(d_{21},d_{22},d_{23},\cdots,d_{2k})$的节点$p_1$和$p_2$来说，它们之间的兴趣相关程度($G$)的计算公式为

作者简介：吴 思(1983－)，男，硕士研究生，主研方向：网络与通信；欧阳松，教授
收稿日期：2007-06-10 **E-mail：**csu_wusi@yahoo.com.cn

$$G = \frac{\sum_{i=1}^{k} d_{1i} \times d_{2i}}{\sqrt{\sum_{i=1}^{k} d_{1i}^2} \times \sqrt{\sum_{i=1}^{k} d_{2i}^2}}$$

其中，$d_{1i} \in \boldsymbol{D}_1$, $d_{2i} \in \boldsymbol{D}_2 (i \in (1,k))$，$k$为兴趣域的长度。

G的数值越大，表示节点p_1和p_2的兴趣相关程度越高，反之表示兴趣相关程度低。在P2P网络中，当节点p做本地搜索找到了查询的资源时，p就与查询发起节点P_{query}建立直接的联系，将资源和自己的兴趣域$\boldsymbol{D}_p$发送给P_{query}，P_{query}在与节点p建立联系后，接收资源和$\boldsymbol{D}_p$，并通过式(1)计算自己和节点p的兴趣相关度G_p，如果$G_p > G_{min}$(G_{min}为P_{query}通过计算自己与当前邻居节点表中每一节点的兴趣相关度的值，并比较得出的最小值)，则节点P_{query}从邻居节点表中删除与G_{min}对应的节点，并将节点p加入到邻居节点表中。该算法表示如下：

算法 1 基于兴趣自组织的查询算法

目的：根据查询在 P2P 网络中搜索资源，同时优化节点的邻居节点表

输入：查询

输出：符合查询条件的资源

在搜索节点方：

```
Optimized_Query(query,TTL){
    If(查询没有重复){
      根据查询进行本地搜索
      If(找到符合查询的资源){
    将资源发送给查询节点
    将自身的兴趣域发送给查询节点 }
TTL 减 1
将查询发送给邻居节点表中的节点
} }
```

在查询发起节点方：

```
GetResourceFromPeer(peer p){
    从节点 p 下载资源
    计算与 p 的兴趣域的相关度 G
    比较G与邻居节点表中节点的最小相关度Gmin
    If(G>Gmin){
    删除邻居节点表中与Gmin对应的节点
    将节点 p 加入到邻居节点表中
} }
```

2 查询转发算法与节点自治

解决广播造成的查询消息洪流的方法之一就是选择性地转发查询，而不是将查询发送到节点的每一个邻居。

2.1 有选择的转发查询

在本文的算法中，节点将每个到来的查询保存在查询历史(QueryHistory)中。当一个查询 q 到来时，节点首先检查查询历史中是否有相同的查询，如果有相同查询，节点将选择性地转发查询 q，如果没有相同查询，节点将广播查询 q。选择进行转发的邻居节点的标准是：设定一个阈值 a，如果某邻居节点兴趣域中与查询对应的项的值大于 a，则转发查询，反之不转发。阈值的设定方法可以根据应用环境的不同而改变，本文实验中的阈值设定为与某兴趣类相关的资源个数。例如：若设定阈值 a=5，查询搜索的是计算机类文档，那么节点在选择性转发的情况下将只向共享计算机类别资源数大于 5 的邻居节点转发。选择性转发查询的查全率比广播查询低，但是本文的转发算法是建立在节点根据兴趣相关度自动聚集的前提下的，因此，转发算法在显著降低查询跳数(减少了网络中的查询消息条数)的同时并没有明显降低查全率。这一特性在实验中也得到了证实。转发算法表示如下：

算法 2 查询转发算法

目的：根据查询历史有选择地转发查询

输入：查询

输出：符合转发条件的节点

```
SendQueryToNeighbors(query,TTL){
      If(查询已经存在于查询历史中){
      在邻居节点中选择兴趣域中与查询类别对应的项的值大于阈值的节点
      转发查询到这些节点}
      Else{ 转发查询到邻居节点表中的所有节点
} }
```

2.2 节点的自治

网络条件包括网络的拥塞与否和邻居节点的工作状态。当网络没有拥塞或邻居节点空闲时，可以采用广播的方式转发查询，以获得较高的查全率。当网络拥塞或者邻居节点忙的时候，可以采用算法 2 选择性地转发查询，这样只会产生少量的查询跳数，并且查全率也不会明显下降。节点获知网络条件的做法是：节点定时向邻居节点发送探测信息(相当于 ping)，检测当前网络的状态，如果探测到邻居节点比较空闲，探测回应(相当于 pong)也在设定的时间内收到，则认为此时网络状态良好，在下一次探测前收到查询时，节点将采用广播的方式转发查询。如果节点探测到邻居节点忙，或者探测回应在设定的时间内没有收到，则认为此时网络状态差，在下一次探测前收到查询时，节点将选择性地转发查询。

3 性能分析

性能分析主要考虑以下代价指标：

(1)存储代价。主要考虑每个节点中邻居节点表的存储代价。假设网络中有N个节点，平均每个节点共有$N_{neighbor}$个邻居节点，邻居节点表中的每一项占用s字节的存储空间，则平均每个节点的存储代价为：$N_{neighbor} \times s$；整个网络的存储代价为：$N \times N_{neighbor} \times s$。由于邻居节点表中的每一项存储了每个邻居的地址和该邻居的兴趣域，因此每个节点的存储代价与邻居节点的个数和兴趣域的长度成正比。在算法 1 中，节点与新节点建立连接时(向表中添加项)，会自动断开(从表中删除项)与当前邻居节点表中兴趣最不相关的节点的连接，所以，$N_{neighbor}$的值不会随查询次数的增多而变大。兴趣域的长度与兴趣类别的数量成正比。兴趣类别划分得越细，查询检索效率越高，但是对存储空间的需求也越大，需要根据实际情况折中。虽然整个网络的存储代价可能高于集中模型中的目录大小，但这些代价被分散到了网络中的所有节点。

(2)网络通信代价。通信代价用需要传输的字节数来衡量。不考虑文件下载的传输量，只考虑查询过程所需的传输量。与Gnutella网络查询过程中的传输量相比，在资源被检索到的情况下，算法 1 增加了$Len_{indomain}$的传输量。$Len_{indomain}$代表节点兴趣域的长度，与本文算法减少的网络中查询消息总量相比，$Len_{indomain}$可以忽略不计。

(3)更新代价。节点通过计算新节点(返回查询结果也不在邻居节点表中的节点)与自己的兴趣相关度，并与当前邻居节点表中最小兴趣的相关度比较，来更新邻居节点表。假设平均每个节点共有$N_{neighbor}$个邻居节点，计算一对节点兴趣相关度的代价为$COST_G$，$COST_G$的时间复杂度为$O(k)$，k为兴趣域中兴趣类别的数量，则平均每个节点的更新代价为$(N_{neighbor}+1) \times COST_G$，从计算公式可以看出更新代价很小。

从以上的分析可以看出，算法 1、算法 2 仅仅增加了少

量代价，基本不会影响网络性能。

4 实验

4.1 实验设置

所有实验都是在一台PC上完成的，实验的原型系统用Java编写。产生的P2P网络拓扑结构基于power-low，用PLOD算法[4]产生。研究显示，Gnutella的网络拓扑结构接近power-low，因此，原型系统接近现实中的非结构化P2P网络。

参数：PLOD算法产生的网络是一个无向图，有 1 000 个节点，节点的平均出度为 3.6，相当于图中有 1 800 条无向边。查询算法的测试文档集采用SMART[5]系统中使用的测试文档集：ADI, CACM, Cranfield collection, CISI。其中，ADI包括 82 个文学类的文档和 35 个查询；CACM包括 3 204 个计算机类文档和 64 个查询；Cranfield collection包括 1 400 个太空动力学文档和 225 个查询；CISI包括 1 460 个信息科学类文档和 112 个查询。所有的测试文档采用 80-20 原则分布在网络的各个节点上(80%的资源分布在 20%的节点上，剩下 20%的资源分布在 80%的节点上，比较接近现实)，文档不会重复出现。

4.2 查询效率和查询效果

为了验证算法的有效性，采用查全率(*Recall*)和查询跳数来衡量，实验比照的对象是比较流行的非结构化 P2P 网络：Gnutella。查全率的计算公式如下：

Recall =查询到的文档数/集中式索引的文档数

比如某查询搜索到 10 个资源，整个 P2P 网络中存在 20 个相关资源，那么某查询的查全率就为 0.5。查询跳数也反映了查询被转发的次数，查询被转发的次数越少，产生的查询消息条数也就越少，有利于保证 P2P 网络的查询性能。

实验 1

称采用搜索算法 1 的 P2P 网络为 UltraGnutella。从图 1 中可以看出，随着 *TTL* 的增大，查全率越来越高，因为查询覆盖的节点数会随着 *TTL* 的增加而增加。

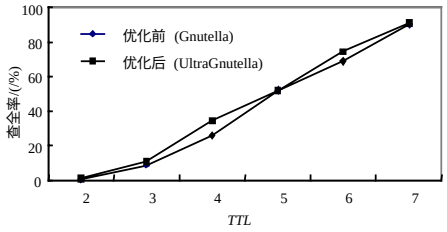

图 1 不同 *TTL* 情况下的相对平均查全率

在实验中，当 *TTL* 取值在 3～6 时，UltraGnutella 的查全率都高于 Gnutella，因为算法 1 将兴趣相近的节点聚集了起来，在 *TTL* 的限制内，查询基本在兴趣相近的节点间转发，搜索到与查询相关资源的几率也较大(作为一个理性的节点，其查询的内容和其兴趣爱好是相关的)。当 *TTL* 取 7 或更大时，两者的查全率基本一致，因为此时查询已经几乎覆盖了网络中的所有节点，查全率几近 100%。在以后的实验中，*TTL* 取值为 5。

实验 2

在采用算法 1 和算法 2 的 UltraGnutella 网络中，节点的查询效率随网络中总查询次数的增加而改善，见图 2。本文选取一个编号为 801 的节点进行测试，该节点的兴趣之一为太空动力学，因此在实验中，801 节点共做了 10 次与太空动力学相关的查询，查询编号为：cran157。每次查询间隔了 20 次随机查询：随机选择网络中一个节点做一次查询，查询内容也是随机的。从图 2 中可以看出，801 节点做第 1 次查询时(*TTL*=5)，查询跳数接近 800 次，因为这是网络中的第 1 次查询，所有节点的查询历史都为空，当节点遇到查询 cran157 时，它们都用广播的方法转发查询，所以第 1 次查询和 Gnutella 网络中的查询是一样的：都有较高的查全率和较多的查询转发次数。当网络中的查询次数累计到 20 次时，801 节点再次查询 cran157。此时可以发现查询跳数降到了 200 以下，而查全率也降低到 0.6 以下。这表明网络中的节点积累了查询历史，遇到相同的查询，将选择性地转发。这也使得查询覆盖的节点比第 1 次查询少，导致查全率的下降。当网络中的查询次数增加，801 节点做 cran157 查询时，查询跳数一直保持在 200 跳左右，查全率也逐步接近 0.8，这是因为查询次数越多，网络中的节点将与自己兴趣相近的节点更紧密地聚集在一起，选择性查询也使查询只在兴趣相近的节点间转发，因此，UltraGnutella 能够在不明显影响查全率的前提下有效避免广播引起的查询洪流。

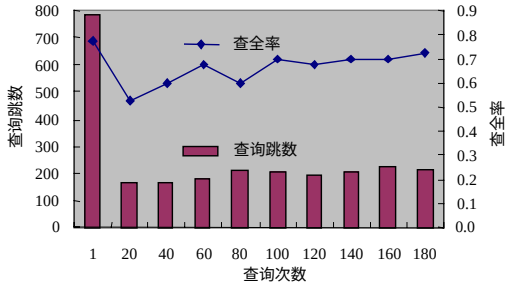

图 2 节点 801 查询 cran157 的查全率和跳数变化

实验 3

实验 3 从网络中随机选出 10 个对太空动力学感兴趣的节点，并在这 10 个节点上做 10 组太空动力学相关的查询，每组查询(每个节点做一次查询)之间间隔 10 次随机查询(随机选择 10 个节点，每个节点做一次查询)。通过计算每组查询的平均查询跳数和平均查全率得出了图 3。

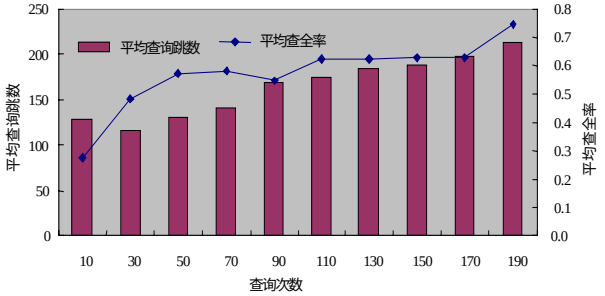

图 3 平均查全率和平均查询跳数随查询次数增加的变化

从图 3 中可以看到：随着查询次数的增加，平均查询跳数和查全率也增加了，而且节点与邻居节点的兴趣相似度越来越高，选择转发查询的对象也越来越多，所以平均查询跳数和平均查全率都增加了。

5 结束语

本文研究了非结构化分散型 P2P 网络中查询检索性能的改进方法。通过引入基于兴趣相关度自组织的查询算法和转

（下转第 107 页）