ISSN 1000-9825, CODEN RUXUEW
*Journal of Software*, Vol.18, No.9, September 2007, pp.2216−2225
DOI: 10.1360/jos182216

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

# 一种P2P环境下基于用户行为的语义检索方案*

邱志欢+, 肖明忠, 代亚非

(北京大学 计算机科学技术系,北京 100871)

# A User Behavior Based Semantic Search Approach under P2P Environment

QIU Zhi-Huan+, XIAO Ming-Zhong, DAI Ya-Fei

(Department of Computer Science and Technology, Peking University, Beijing 100871, China)

+ Corresponding author: Phn: +86-10-62751799 ext 8013, E-mail: qzh@net.pku.edu.cn, http://net.pku.edu.cn

**Qiu ZH, Xiao MZ, Dai YF. A user behavior based semantic search approach under P2P environment. *Journal of Software*, 2007,18(9):2216−2225.** http://www.jos.org.cn/1000-9825/18/2216.htm

**Abstract**: Restricted by the diversity of resources and the complexity of search algorithms, current search mechanisms in peer-to-peer file sharing systems are based on file names and simple keyword matching. These mechanisms cannot recognize deeper relationships between keywords and resources; hence it cannot provide high search quality. This paper proposes a new search scheme, which is built on top of the current peer-to-peer network. It harnesses users' search behaviors and download behaviors to automatically discover the deeper relationships between keywords and resources, which is then used to improve the search quality. It has the advantages of low implementation cost, low complexity, self-evolving, and supports for semantic search. Simulations based on the Maze system show that this approach has high search hit rate and accuracy.

**Key words**: P2P; peer-to-peer network; user behavior; semantic search; data mining; Maze

**摘 要**: 受资源类型多样化、搜索复杂度的制约,现有的 P2P 文件共享系统中的搜索机制是基于文件名的关键字匹配,这种方法不能发现关键字与资源内容之间的深层关系,因此不能实现语义检索.针对这个问题,提出一种新的搜索方案,该方案建立在已有的搜索机制之上,利用用户的搜索行为和下载行为的规律自动发现关键字和资源间的深层关系,在底层的 P2P 网络上构建一个元数据空间以辅助搜索.该方案具有实现代价小、时间复杂度低、可进化和支持语义搜索的优点.在 Maze 系统上的实验表明,该方案具有较高的查询命中率和查询准确率.

**关键词**: P2P;对等网络;用户行为;语义检索;数据挖掘;Maze

**中图法分类号**: TP393 **文献标识码**: A

## 1 问题的提出

现有的 P2P 文件共享系统中的搜索是基于文件名的关键字搜索,搜索能够返回结果,当且仅当系统能够找到文件名和关键字匹配的资源.在传统环境下,文件名的长度一般较短,所含的信息量较少,但由于 P2P 文件共享

* Supported by the National Natural Science Foundation of China under Grant No.90412008 (国家自然科学基金); the National Basic Research Program of China under Grant No.2004CB318204 (国家重点基础研究发展计划(973))

Received 2006-07-03; Accepted 2006-09-30

系统通常都存在着激励机制,使得用户具有共享文件的动力,他们把一些新的关键字加入到文件名中以增加文件被搜索到的概率,从而使其所包含的信息量也相应地增加.实践表明,这种关键字匹配方法简单、有效,但它毕竟是较浅层的字符串匹配,不能根据文件内容做更有语义的搜索.

Web 搜索引擎使用非常成熟的文本检索技术,在网页搜索方面已经有很好的效果.但为何 P2P 搜索还未能达到这一程度呢?探究其原因,有两方面:(1) Web 搜索引擎将索引信息存储在一个封闭可控的环境中,因此可以快速地利用它们,而后者是开放性的,结点的加入和退出频繁,要存储和访问信息相对比较困难;(2) Web 搜索引擎的处理对象主要是文本,相对容易处理,而后者的资源是多样化的,其中包括各种多媒体资源,针对这些资源的基于内容的检索技术目前还不够成熟,很难用机器自动处理的方式给用户提供一个令人满意的效果.

目前,在 P2P 语义检索方面的研究主要集中在以下几种方法:(1) 利用兴趣局部性(interest-based locality)提高检索性能:文献[1]基于这一原理,提出使用兴趣捷径(interest-based shortcuts)来改进 Gnutella 的查询性能的方案,该方案通过结点的查询历史或与其他结点交流的方式来获得感兴趣的结点地址作为捷径,在以后的查询中优先向这些结点发送查询请求.实验表明,这种方案可以大量减少 flooding 的数量;(2) 利用用户的兴趣分组提高检索性能:文献[2]认为用户的兴趣是比较稳定的,即结点未来查询的数据和结点当前存储的数据应该同属于一个类别,它通过计算结点上所有文件名的词频向量的 Jensen-Shannon 分歧值来得到结点的相似度,进而将用户进行分组,在以后的查询中,用户向比较有可能返回结果的用户组发出查询.文献[3,4]也提出了用户分组的思路,但文献[3]利用资源的类别得到用户的兴趣,而文献[4]则通过分析结点的历史查询记录发现用户的兴趣;(3) 在 P2P 网络中部署本体(pntology):文献[5]在 P2P 网络中构建一个基于 RDF 的元数据空间,并解决分布式查询和概念间的合并、映射、进化及资源的注解等问题.这些方法有一个共性:它们不解决元数据的生成问题,如果用户不提供元数据,那么算法只能回退到使用基于文件名的关键字匹配方法.

针对上述问题,一个更好的 P2P 检索系统应该满足下面 3 个要求:1) 搜索响应时间短;2) 返回的结果质量高:能够深层地挖掘关键字和资源内容的关系;3) 可进化:能够随着用户的使用自适应地改进自身的搜索性能.

本文的核心思想是:P2P 文件共享系统中存在着海量的搜索和下载行为,通过分析这些行为可以发现关键字和资源相关性的大小——相关度,系统可以利用相关度来进行搜索.由于利用了人对资源的理解来发现相关性,我们称其为语义检索,其特点是:(1) 能够发现关键字和资源内容的相关性;(2) 随着时间的推移,系统根据用户行为自动调整相关度;(3) 它的工作方式是对系统原有的搜索机制进行扩展而不是取代,同时对原有搜索机制的要求并不苛刻,两者间相对比较独立,因而适用性广.这一思想包含两个部分:第一部分是如何发现搜索和资源的相关性;另一部分是如何利用相关性进行搜索.这将在本文的第 2 节中加以描述,包括系统概述、相关性的定义和算法等.第 3 节用模拟实验对本方案进行验证.第 4 节总结全文并讨论下一步的工作.

## 2 系统模型

### 2.1 概 述

在本方案中,搜索和资源之间的相关度被保存在一个称为 MetaSpace 的抽象元数据空间里,它构建在底层 P2P 网络之上,两者之间的关系如图 1 所示.系统开始运行时,MetaSpace 不包含任何数据,这时,用户提交的搜索全部由底层系统原有的搜索机制完成.在系统的运行过程中,每个结点将本地用户的搜索和下载行为记录到一个缓冲区中,经过一段时间后,对这些行为批量地进行分析,生成关键字和资源的对应关系,然后将这些对应关系发布到 MetaSpace 中.MetaSpace 收到结点发布的对应关系时,计算它们的相关度并保存下来.

当 MetaSpace 形成一定规模时,用户提交的搜索关键字便会以一定的概率在其中找到与之匹配的对应关系,根据相关度将这些对应关系进行排序,便可以得到一个有序的资源列表,系统将这个资源列表作为搜索结果返回给用户.若没有用户想要的结果,系统将关键字提交给底层系统原有的搜索机制继续进行搜索.

### 2.2 对底层系统的要求

为了存储和处理自动生成的元数据,本方案对底层的P2P网络有一定的要求.当前存在着多种P2P网络,本方

案面向的是基于分布式哈希表DHT[6−8]的结构化P2P网络,而不是基于Flooding[9],RandomWalk[10]等机制的非结构化网络.另外,不失一般性,我们假设底层P2P网络支持从“资源内容标识”(基于资源内容生成的id,如MD5 摘要)到“资源地址”的查找功能.当前,同时满足这两个条件的P2P文件共享系统是存在的,比如新版的eMule[11]和BitTorrent[12]等等.

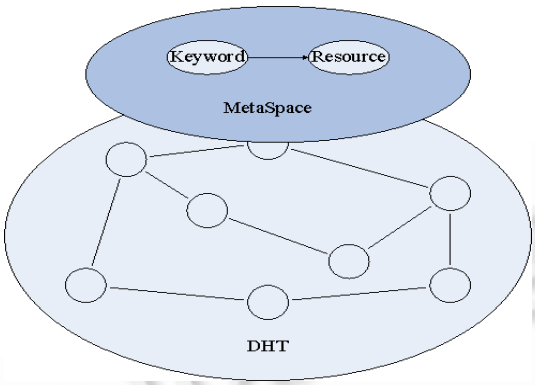



Fig.1 System architecture

图 1 系统结构

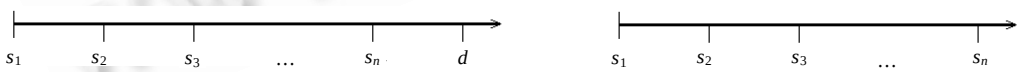
### 2.3 MetaSpace的定义

MetaSpace 的定义为 *MetaSpace*={⟨*k*,*cid*,*c*⟩}.其中,*k* 是关键字,*cid* 是资源的内容标识,*c* 是 *k* 和 *cid* 的相关度.为了将 MetaSpace 这个集合存储到 P2P 网络中,集合的每一个元素将以 *k* 为键、以(*cid*,*c*)为值的方式发布到底层 DHT 网络中.

### 2.4 MetaSpace的生成

MetaSpace中的数据从何而来?这是一个元数据的生成问题,解决这个问题有两种基本途径[13]:手动生成和机器自动生成.手动生成的元数据比较准确,然而却存在代价大、不灵活的缺点.自动生成方法依赖于机器学习和自然语言处理技术,代价小而且灵活.虽然当前机器智能的方法还不够成熟,但在实践中特别是文本检索领域中的一些自动生成算法已经能够达到不错的效果.然而,P2P共享系统中的资源是多样化的,根据一项调查[14],在Maze这个P2P文件共享系统中,音视频和图像占了文件总数的 39%及存储量的 79%.另外,系统中还存在着压缩格式、可执行格式等各种非文本格式的资源.处理这些数据将是一件非常困难的事情.

是否可以从其他角度出发来解决这个问题?考虑如下场景:用户*A*想要下载某一个资源,因此发出一个搜索请求,若系统没有返回让*A*满意的结果,则*A*更改关键字作下一次搜索;若有,则停止搜索.这样重复几次,直到系统返回*A*所需要的结果或者*A*的耐性丧失为止,从而形成一个操作序列($s_1$,$s_2$,…,$s_n$,*d*),或者是($s_1$,$s_2$,…,$s_n$),如图 2 所示(其中,$s_i$表示第*i*次搜索,*d*表示下载).




(a) Desired resource located (a) 成功找到所需资源

(b) Desired resource not located (b) 未找到所需资源

Fig.2 An operation sequence of user *A*

图 2 用户 *A* 的一次操作序列

分析这个过程,在*A*看来,他所提交的若干个关键字和他所要下载的资源都是相关的,只是由于系统的检索机制不能及时发现这种相关性,因此导致要进行多次搜索才能得到所需结果.若利用这种相关性,例如在$s_1$和*d*之间建立起直接关联,则可以改善系统的检索性能.因为这种相关性发生在较短的时间间隔之内,以下称其为“短期相关性”.MetaSpace的自动生成的核心思想就是利用短期相关性.下面给出它的严格数学定义.设系统的

用户集合为$U$,对于任一个$u\in U$,它的下载行为是$D_u$,搜索行为是$S_u$.其中,对于任意的$d\in D_u$, $d=\{cid\}$;对于任意的$s\in S_u$,$s=\{k\}$.

**定义 1(短期相关)**. 给定用户$u\in U$,对于任意的行为$a,b\in D_u\cup S_u$,设$a,b$的发生时间分别为$t_a$和$t_b$,若$|t_a-t_b|\le t_h$,则称$a$和$b$是短期相关的,记为$T(a,b)$.其中,$t_h$为一个正实数常数.

**定义 2(相关关系)**. 给定用户$u\in U$,对于任意的行为$s\in S_u$和$d\in D_u$,$s$和$d$是相关的当且仅当下面两个条件之一成立,记为$R(s,d)$:

1) $T(s,d)$;

2) 存在$s'\in S_u$,使得$T(s,s')$和$R(s',d)$同时成立.

称 $k$ 和 $cid$ 是相关的,记为 $R'(k,cid)$,当且仅当存在着 $s$ 和 $d$,使得 $k\in s,cid\in d$,且 $R(s,d)$成立.

在理想的情况下,对于用户的每一个下载操作,都有一个精确的与之相关的搜索序列.但在实际情况中,用户往往会并发地进行多个目标的搜索,这就使得呈现出来的操作序列是由多个序列重叠在一起而得到的,由定义 2 定义的相关关系就可能会使本来不相关的搜索行为和下载行为变得相关,从而出现噪声.需要用某种方式来消除噪声,因此,为每一个$\langle k,cid\rangle\in R'$都赋予一个引用计数 $c$,其作用将在下一节加以介绍.

**定义 3(引用计数)**. $\langle k,cid\rangle\in R'$的引用计数 $c$ 是一个非负整数,其值为使 $R'(k,cid)$成立的操作序列个数.

通过简单的计算,可以得到系统中单个结点的 $R'$及 $R'$中元素的引用计数,汇总各个结点的信息,就得到系统整体的元数据空间 MetaSpace.下面给出 MetaSpace 的生成算法.

**算法 1**. 发布 $R'$和相应的引用计数集.

输入:一段时间内的搜索记录 $S$ 和下载记录 $D$.

(1) 将 $S$ 和 $D$ 的元素按时间的先后顺序排列;

(2) 计算 $R'$和相应的引用计数集;

(3) 对于任意的$\langle k,cid\rangle\in R'$和相应的 $c$,以 $k$ 为键,$(cid,c)$为值,发布 $k\rightarrow(cid,c)$到 DHT 网络中.

当结点收集到一定数量的用户行为之后,便执行算法 1 进行批量分析,然后将结果发布到系统中去.设 $R'$中不同的 $k$ 的个数为 $m$,结点数为 $n$,由于 DHT 网络定位结点的时间复杂度为 $O(\log n)$,若按顺序线性地定位 $m$ 个结点,则算法 1 的时间复杂度为 $O(m\log n)$;若并发地定位 $m$ 个结点,则时间复杂度为 $O(\log n)$.

**算法 2**. 整合 $R'$和引用计数集.

输入:$k\rightarrow(cid,c)$.

(1) 在本地查找是否存在 $k\rightarrow cid$ 的映射;

(2) 若存在,将原来的引用计数加上 $c$,转到第(5)步;

(3) 检查本地缓存空间是否已经达到一定的上限,若超出,则按 LRU 的原则淘汰一个映射;

(4) 在本地新建一个映射 $k\rightarrow cid$,并将它的引用计数设为 $c$;

(5) 算法结束.

当结点收到算法 1 发布过来的信息时,执行算法 2,将原有的信息和新得到的信息进行整合.为了防止结点存储空间耗费过大,每个结点设置一个空间上限,并用 LRU 法则淘汰映射.算法 2 是一个本地的算法,在执行过程中无须访问其他结点的信息,它的空间复杂度为 $O(1)$,时间复杂度取决于第(1)步和第(4)步,在使用哈希表的情况下,平均时间复杂度为 $O(1)$.

### 2.5 使用MetaSpace进行搜索

MetaSpace生成后,可以从中得到的信息为形如$\langle k,cid,c\rangle$这样的三元组.搜索问题实际上是一个数据挖掘中的布尔关联规则挖掘问题[15],可以被描述为:当用户提交一组关键字$\{k\}$时,他很可能会下载什么的资源$cid$,下载它们的可能性有多大.关联规则挖掘中的核心概念是支持度和置信度,为了根据MetaSpace计算这两个参数,首先定义单个关键字到单个资源$k\Rightarrow cid$的支持度和置信度.

**定义 4(支持度)**. 对于任意的$k$和$cid$,若$\langle k,cid,c\rangle\in MetaSpace$,则$k\Rightarrow cid$的支持度为$c$,否则为 0.将它记为$Support(k\Rightarrow cid)$.相应的最小支持度记为$S_{\min}$.

**定义 5(置信度)**. 对于任意的$k$和$cid$,设$c_k=\sum_{\langle k,id,c\rangle\in MetaSpace} c$,若$\langle k,cid,c\rangle\in MetaSpace$,则$k\Rightarrow cid$的支持度为$c/c_k$,否则为 0.将这个值计为 $Confidence(k\Rightarrow cid)$.

这两个参数的计算是由 DHT 网络中负责关键字 $k$ 的结点来完成的,当它收到关于 $k$ 的查询请求时,只需根据本地的信息就可以完成计算.下面定义$\{k\}\Rightarrow cid$ 的相关度.

**定义 6(查询相关度)**.对于任意的$\{k\}$和 $cid$,$k\Rightarrow cid$ 的相关度记为 $Sim(\{k\}\Rightarrow cid)$,其值为

$$Sim(\{k\}\Rightarrow cid)=\sum_{k'\in\{k\}\wedge Support(k',cid)\geq S_{\min}} Confidence(k',cid).$$

当用户提交一个查询 $K=\{k\}$时,利用 MetaSpace,系统的查询算法如下:

**算法 3**. 根据关键字集合 $K=\{k\}$查询资源.

(1) $V\leftarrow\varnothing$.

(2) 对于每一个 $k\in K$.

(3) 向底层DHT网络发出请求,获得满足$Support(k\Rightarrow cid)\geq S_{\min}$的三元组集合$V_k=\{\langle k,cid,Confidence(k\Rightarrow cid)\rangle\}$,令$V=V\cup V_k$.为了防止网络流量过大,发送方给$|V_k|$设置一个上限$v_{\max}$,若$|V_k|>v_{\max}$,则将$V_k$中$Confidence(k\Rightarrow cid)$最小的三元组淘汰出去,直到$|V_k|\leq v_{\max}$为止,然后再将$V_k$传输出去.

(4) 若 $V=\varnothing$,转到第(8)步.

(5) 令 $C=\{cid|\exists k\rightarrow\langle k,cid,Confidence(k\Rightarrow cid)\rangle\in V\}$,对于每一个 $cid\in C$.

(6) 根据 $V$ 计算 $Sim(\{k\}\Rightarrow cid)$.

(7) 将 $C$ 中的元素按 $Sim(\{k\}\Rightarrow cid)$从大到小的顺序呈现给用户.转到第(9)步.

(8) 将$\{k\}$交给系统原有的搜索功能进行搜索.

(9) 算法结束.

设系统中的结点数为$n$,$m=|K|$,采用并发的方法,则算法第(2)步和第(3)步的执行时间为$O(\log n)$的定位时间加上$O(mv_{\max})$的传输时间,第(5)步和第(6)步的时间复杂度为$O(mv_{\max})$,第(7)步的执行时间为$O(mv_{\max}\log(mv_{\max}))$.如果我们只关心网络传输时间,则算法 3 的时间复杂度为$O(mv_{\max}+\log n)$;如果把本地执行时间考虑进去,则时间复杂度为$O(mv_{\max}\log(mv_{\max})+\log n)$.

## 3 实验分析

为了检验上述方案的性能,我们利用实际 P2P 应用系统中的搜索和下载日志,模拟 MetaSpace 的生成过程和搜索过程,对 MetaSpace 的各项性能参数、搜索的命中率和准确率进行评测.实验环境为 Redhat EL AS4 操作系统,Intel Xeon MP 1.90G CPU×8 16G 内存.

### 3.1 数据集

本文模拟实验的数据来自于 Maze 系统,采用了从 2005 年 1 月 1 日~2005 年 4 月 28 日共 116 天的搜索和下载记录(中间有若干天由于服务器故障使得日志丢失),共计 13 478 365 条搜索记录和 87 054 774 条下载记录.可以看到,平均每一次搜索就有 6.46 个文件被下载,这有几个原因:首先,部分文件并不是搜索到的,用户可以通过浏览共享目录发现自己感兴趣的资源,经过统计发现约有 52%的下载文件是搜索到的;其次是资源的粒度问题,一些资源本身就由好几个文件组成,比如常见的 divx 格式的电影就一般会由若干个 avi 和字幕文件组成.如果我们将同一个用户从同一个其他用户在同一时间下载的所有文件认为是同一个资源,那么事实上,在这 116 天的时间里,只有 15 680 537 个资源被下载,平均是 5.55 个文件/资源,再乘上 52%的比重,可以得到平均情况下为 13 478365/(15 680 537×52%)=1.65 次搜索/资源.

### 3.2 实验结果

#### 3.2.1 资源的下载量

为了在搜索时 MetaSpace 能够起到作用,算法收集的数据必须有一定的支持度,即对同一资源的下载次数

必须达到一定的数目以上.对各个月份进行分析,令 $D(n)$表示当月所有被下载了 $n$ 次的文件的总下载次数占当月所有文件的总下载次数的比例,随着 $n$ 的变化,$D(n)$的变化规律如图 3 所示.

图 3 是关于 $D(n)$的累积分布图,可以看到这几个月 $D(n)$的分布都很接近,有超过 60%的下载量集中在下载次数大于或等于 10 的文件上,超过 30%的下载量集中在下载次数大于或等于 100 的文件上.这意味着,在这几个月份中随机选取一个下载行为,它下载一个当月被下载 10 次或以上的文件的可能性大于 60%,下载一个当月被下载 100 次或以上的文件的可能性大于 30%.根据前面 1.65 次搜索/资源的结论,这说明,我们有较大的可能性为用户当前想要下载的文件收集足够多的搜索行为,从而构造具有一定支持度的元数据.

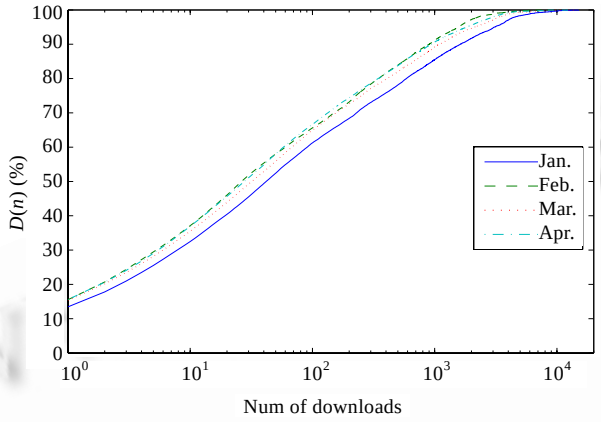



Fig.3 Cumulative distribution graph of $D(n)$

图 3 $D(n)$的累积分布图

3.2.2 MetaSpace 的生成

MetaSpace自动生成的基础是短期相关性,而短期相关性的定义又依赖于时间界限$t_h$的大小.$t_h$应该如何取值?$t_h$的变化会引起系统在 3 个方面发生变化:(1) MetaSpace的大小:“有效下载”的个数,即能够找到一个短期相关搜索行为的下载行为的个数,会随着$t_h$的增大而增多,从而使得MetaSpace变大,最终使得结点的空间耗费变大;(2) “无效下载”的个数,即找不到短期相关搜索行为的下载行为,会随着$t_h$的增大而减小;(3) 短期相关的准确率,即关系$T$和实际情况的接近程度,会随着$t_h$的变化而变化,当$t_h$过大时,一些本不相关的行为变得相关,而当$t_h$过小时,一些本来是相关的行为却会被认为是不相关.由于算法采用支持度和置信度对数据进行筛选,因$t_h$过大而带来的那些“偶然相关”的行为将会由于得不到足够的支持度和置信度而被淘汰,不会对算法的正确率产生太大的影响,即对$t_h$值偏大的选择不会对算法的正确率产生太大的影响,这一点将在下一节给予说明.本节关注$t_h$对MetaSpace的大小和“无效下载”个数的影响.

可以预计,当$t_h$小到一定程度时,所有的行为都不相关,这时三元组的数量为 0;而当$t_h$大到一定程度时,同一个用户的所有行为都相关,这时三元组的数量达到最多.因此,MetaSpace的大小是一个上下有界的量.对 2005 年 1 月~4 月的数据进行分析,MetaSpace的大小与$t_h$的关系如图 4、图 5 所示.图 4 说明了这种趋势,随着$t_h$的增大,MetaSpace的增长速度越来越慢,最后将趋向于一个常数.从图 4 中可以发现,各个月份的元数据空间大小差别比较大,这实际上是因为各个月份的用户数量差别比较大.根据文献[15]的调查,Maze系统中有 0.47%的用户有 1/3 以上的时间在线并具有很强的服务器性质,我们保守地估计用户的 1/1000 能够构成一个比较稳定的P2P网络,将MetaSpace的大小除以这些用户的数目,得到平均每个结点上的空间耗费,如图 5 所示.这时,各月份的结点空间耗费差别变小,可见,MetaSpace的大小与用户数量是成一定比例关系的.

经过统计,在Maze系统中,用户搜索的关键字长度平均为 8.2 字节,内容标识若采用MD5 摘要,则长度为 16 字节,加上 8 字节的引用计数, MetaSpace中每一个三元组占据的空间为 8.2+16+8=32.2 字节,考虑到维护数据结构需要额外的空间耗费,假设一个三元组占据 50 字节的空间.如果要求系统中一个结点使用的缓存空间不能超

过 10M,则每一个结点上的三元组数量不能超过 $2\times10^5$个.对照图 5,当$t_h$<15 时,可以满足这个要求.

在P2P系统中,用户要找到一个资源可以通过两种方式:1) 通过搜索;2) 通过其他方式,如浏览共享目录、系统推荐下载等等.从而,根据定位资源的方式,用户的下载行为相应地也可以分成两类.对Maze的用户行为进行分析可以发现,第 1 种下载行为占了约 52%的比例.因此,一个合理的$t_h$值应当使得“有效下载”的比例接近 52%.图 6 说明了这两者之间的关系.从中可以看出,当$t_h$=15 时,有效下载的比例为 50%左右.

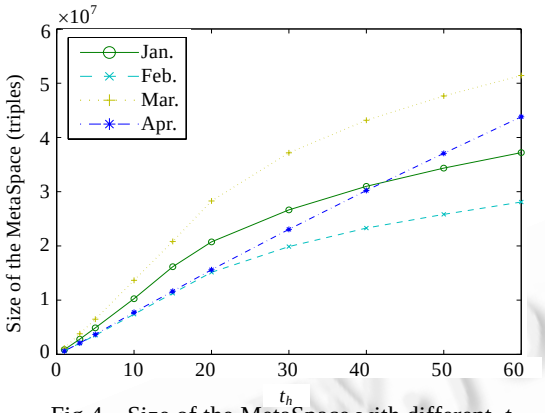



Fig.4 Size of the MetaSpace with different $t_h$

图 4 不同$t_h$取值下MetaSpace的大小

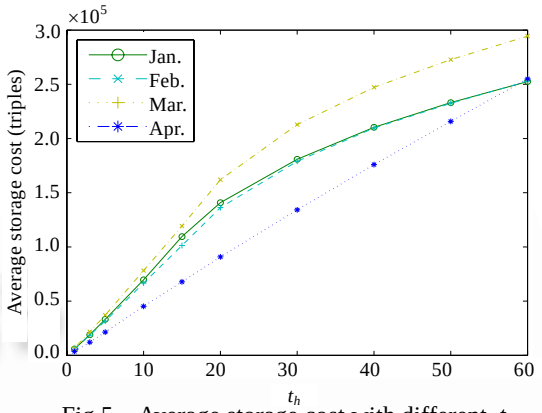



Fig.5 Average storage cost with different $t_h$

图 5 不同$t_h$取值下结点的平均空间耗费

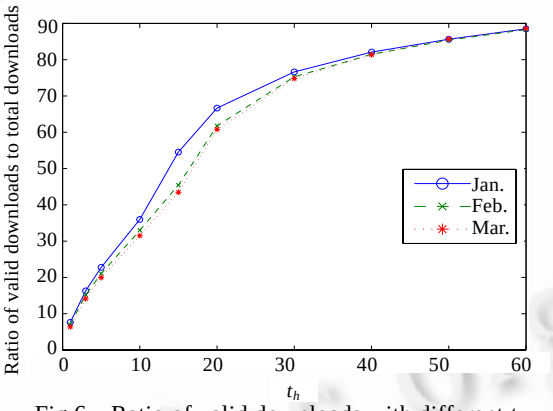



Fig.6 Ratio of valid downloads with different $t_h$

图 6 在不同的$t_h$值下有效下载所占的比重

综合上面的考虑,$t_h$=15 是一种较好的选择.

### 3.2.3 使用 MetaSpace 进行搜索

本节对搜索算法进行实验分析,搜索算法的性能用下列参数来衡量:

(1) 命中率:即用户给出一个搜索,算法能够在 MetaSpace 中找到结果的概率.其计算公式为

$$命中率=\frac{能在MetaSpace中找到结果的搜索数}{总搜索数}.$$

(2) 准确率:即算法在 MetaSpace 中找到的结果满足用户需求的概率.其计算公式为

$$准确率=\frac{在MetaSpace中找到的结果满足用户需求的搜索数}{能在MetaSpace中找到结果的搜索数}.$$

影响算法执行的主要参数有:(1) 短期相关时限$t_h$.由上一节分析可知,$t_h$的变化会影响短期相关的准确性,从而影响搜索算法的命中率和准确率,但可以预计,当$t_h$在一个合理的范围内变动时,算法不会受到太大的影响.(2) 最小支持度$S_{\min}$.其取值同样会影响算法的命中率和准确率,如何对其进行取值则需要进一步的考虑.

(3) $v_{max}$.其过大会造成网络流量过大,过小则可能会将有用的三元组淘汰出去,造成搜索的准确率下降.

图 7 给出了在不同$t_h$值下的MetaSpace的查询命中率和原有系统的有效查询率.这里的有效查询率被定义为能够找到相关下载行为的查询所占的比例,由于它将和一个下载相关的所有搜索都被认为是有效的,而实际上这些搜索中一般只有最后一个搜索是能返回所需结果的搜索,因此,它是原有系统的查询准确率的一个上界.在图 7 中可以看到,如前面所预测,在 2 月份和 4 月份,当$t_h$大于 20 时,查询命中率已经趋向稳定,其大小为20%~30%之间.这个值偏小,但是底层系统本身的有效查询率就比较小,为了衡量它们的相对大小,将这两个值相除求出比值,得到的结果如图8所示.从中可以看到,在2月份,MetaSpace的查询命中率为底层系统的有效查询率的 50%以上;而到了 4 月份,查询命中率基本保持在有效查询率的 1 倍以上.

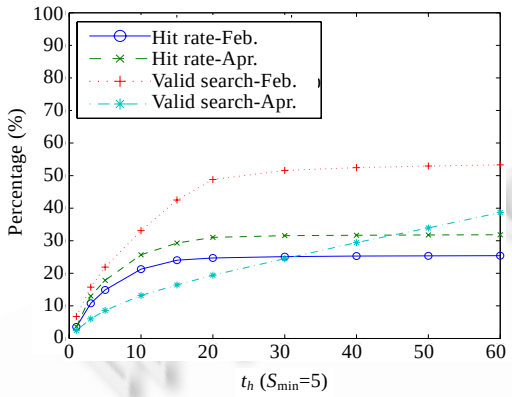



Fig.7 Hit rate and valid search rate with different $t_h$

图 7 在不同$t_h$值下的命中率和有效查询率

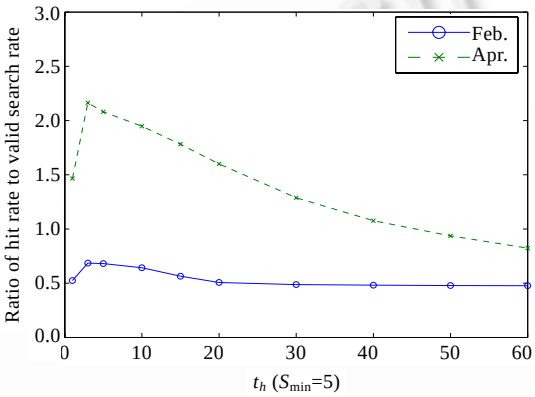



Fig.8 Ratio of hit rate to valid search rate with different $t_h$

图 8 在不同$t_h$值下的命中率和有效查询率的比值

命中率和$S_{min}$之间是什么样的关系?取$t_h$=15,图 9 给出了在不同$S_{min}$值下命中率的变化曲线.命中率受$S_{min}$的影响比较大,通过适当地选择$S_{min}$的值,可将命中率保持在较高的水平.从图 9 中可以看出,当$S_{min}$≤10 时,命中率可以保持在 20%以上.基于同样的考虑,将命中率和有效查询率相除,得到图 10.从中可以观察到,当$S_{min}$≤10 时,两者的比值在 2 月份可保持在 0.5 以上,而到了 4 月份则可保持在 1.5 以上.

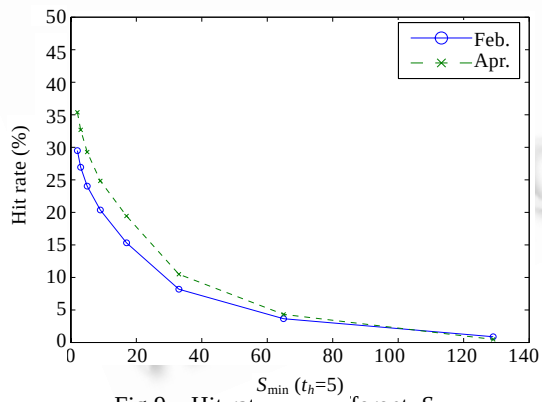



Fig.9 Hit rate with different $S_{min}$

图 9 不同$S_{min}$值下的命中率

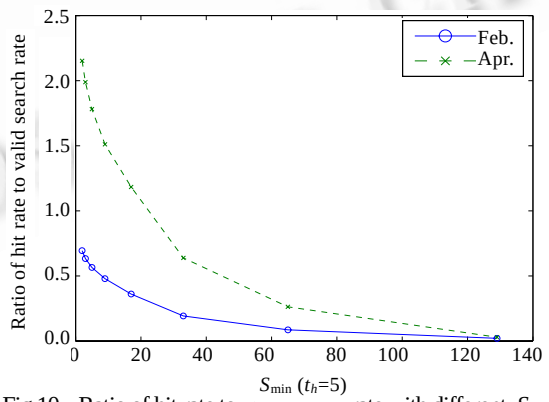



Fig.10 Ratio of hit rate to valid search rate with different $S_{min}$

图 10 不同$S_{min}$值下的命中率和有效查询率的比值

根据上述实验分析,适当地选择$t_h$和$S_{min}$可以得到较高的命中率.那么,命中后的那些搜索的准确率又如何?为了对这个问题进行量化的评测,需要知道用户提交一个搜索时想要下载的资源,途径有:(1) 在实际的系统中运行算法,收集用户的行为;(2) 将测试集分发给志愿者进行测试并收集反馈;(3) 分析原有系统的日志,发现用户提交关键字时想要下载的资源.第 1 种是最好的方案,但是要对系统进行修改,耗费很大;第 2 种方案则很难收集到足够多的反馈.因此,本文采用第 3 种方案.第 3 种方案的难度在于:日志中存在着大量的噪声,难以确定用户

在发出一个搜索时真正想要下载的资源.为此,还是利用多用户的行为来消除这种噪声.当一个搜索和一个资源的相关度达到一定程度以上时,就认为用户在给出该搜索时确实想要下载该资源.在下面的实验中,利用 2~4 月份的数据生成 3 个搜索和资源的对应关系集(相关度大于或等于 10 时认为它们是实际相关的),然后用这 3 个对应关系集分别检验 1~3 月份的MetaSpace,得到的结果如图 11、图 12 所示.

搜索准确度主要受两个参数的影响:$S_{\min}$和$v_{\max}$.$S_{\min}$的作用是将支持度低的三元组淘汰以消除噪声,但是从图 9 和图 11 中看到,$S_{\min}$增大使命中率和准确率都下降,即它的值是越小越好.分析其原因,在搜索算法的第 3 步中,当$S_{\min}$的值增大时,会使可选的三元组数量变小,从而使返回的结果数量变小,因此一些对用户有用的三元组被淘汰出去了,最终使得准确率降低.实验说明,我们可以将$S_{\min}$设成一个较小的值.

图 12 给出了搜索准确度随$v_{\max}$变化的曲线,可以看出,$v_{\max}$越大,准确度的增长越慢,当$v_{\max}$>200 时,准确度变化的幅度就已经非常小了.若取$v_{\max}$=200,我们可以获得的准确度为 70%左右.实际上,由于在P2P共享系统中占大部分的多媒体资源都会存在差别不大的不同版本,而用户却不会太在意下载到的是哪一个版本,上述评测没有考虑到这一点,因此在实际应用中准确率应该会更高.可以换一个角度来观察图 12,将横轴作为搜索结果中呈现给用户的资源的排名,将纵轴看成是准确度的累积分布,可以发现,排名越靠前的资源越有可能是用户想要的资源.事实上,用户在排名前 10 的资源中找到目标的概率是 37%,而在排名第 11~第 20 间找到目标的概率是 10%.

通过上面的实验分析可知,当$t_h$=15,$S_{\min}$=5 及$v_{\max}$=200 时,查询命中率为原有系统有效查询率的 0.7 倍以上,最高可达 2 倍,而命中情况下的准确率为 75%以上,最高可达 80%.由于有效查询率是原有系统查询准确率的一个上界,因此可以推知,MetaSpace的查询准确率最低是原有系统的查询准确率的 0.7×75%=52.5%以上,最高为 2×80%=160%以上.与原有的搜索机制相比,一方面保持了查询准确度,另一方面也实现了语义搜索.

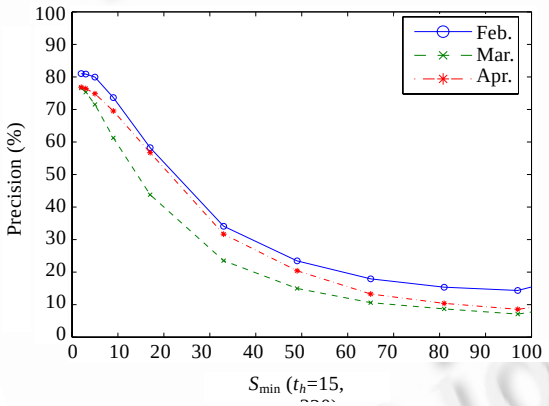



Fig.11 Search precision with different $S_{\min}$

图 11 不同$S_{\min}$值下的搜索准确度

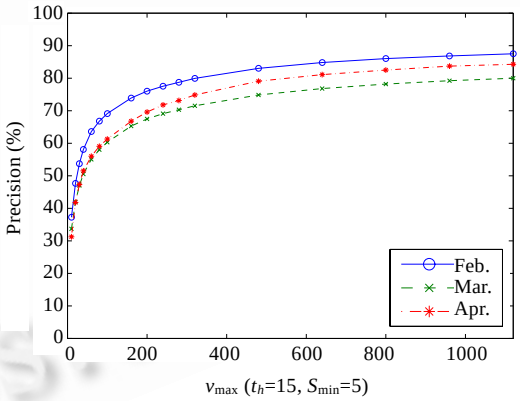



Fig.12 Search precision with different $v_{\max}$

图 12 不同$v_{\max}$值下的搜索准确度

## 4 总结和后续工作

本文提出了一种新颖的语义搜索方案,它利用 P2P 共享系统中海量用户的搜索行为与下载行为自动计算关键字与资源内容的相关度,利用相关度来实现基于资源内容的搜索机制,具有实现代价小、时间复杂度低、可进化和支持语义搜索的优点.文章通过模拟实验证明了它具有较高的查询命中率和准确率.虽然实验是利用 Maze 系统的搜索和下载日志进行的,但是对于终端用户而言,底层系统采用的是何种机制是透明的,因而对于 P2P 共享系统这一类系统来说,用户的搜索和下载行为应当是类似的.因此,实验的结果具有一定的普适性.

本方案除了上述的用途以外,对算法作稍微的改动,还可以用于 P2P 环境下的下载推荐功能,即系统在用户发出查询之后,将与其相关的资源按置信度从高到低的顺序推荐给用户.由于关键字和资源的相关度是通过海量用户的行为来发现的,可以推知这种推荐机制是合理的.

下一步的工作包括:算法的几个关键参数在本文中是通过实验对系统日志进行分析后确定下来的,而最佳的方法应该是让系统根据实际运行情况进行自适应的调整,如何设计自适应的调整算法是要进一步考虑的问题;由于噪声的存在,本文试图从支持度和置信度两个角度出发来确定搜索和下载的相关性,是否可以用更加精确的方法来发现这两者间的相关性,这也是可以进一步考虑的问题.

**致谢** 杨懋博士和徐泉清同志对本文提出了宝贵的意见,作者在此对他们表示感谢.

**References**:

[1] 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. IEEE Press, 2003. 2166−2176.

[2] Asvanund A, Krishnan R, Smith M, Telang R. Interest-Based self-organizing peer-to-peer networks: A club economics approach. In: Proc. of the 13th Workshop on Information Technology and Systems. 2003. http://www.business.uconn.edu/users/atung/seminar/fall2004/smith-paper.pdf

[3] Crespo A, Garcia-Molina H. Semantic overlay networks for P2P systems. In: Moro G, Bergamaschi S, Aberer K, eds. Proc. of the 3rd Int'l Workshop on Agents and Peer-to-Peer Computing. Berlin: Springer-Verlag, 2004. 1−13.

[4] Singh S, Ramabhadran S, Baboescu F, Snoeren AC. The case for service provider deployment of super-peers in P2P networks. In: Proc. of the Workshop on Economics of P2P Systems. Berkeley, 2003. http://www.cs.ucsd.edu/~susingh/papers/tbs-p2pecon03.pdf

[5] Nejdl W, Wolf B, Qu C, Decker B, Sintek M, Naeve A, Nilsson M, Palmer M, Risch T. EDUTELLA: A P2P networking infrastructure based on RDF. In: Proc. of the 11th Int'l World Wide Web Conf. IEEE Press, 2002. 604−615.

[6] Stoica I, Morris R, Karger D, Kaashoek MF, Balakrishnan H. Chord: A scalable peer-to-peer lookup service for Internet applications. In: Proc. of the ACM SIGCOMM 2001. San Diego: ACM Press, 2001. 149−160.

[7] Zhao B, Kubiatowicz J, Joseph A. Tapestry: An infrastructure for fault-tolerant wide-area location and routing. Technical Report, UCB/CSD-01-1141, Computer Science Division, U. C. Berkeley, 2001.

[8] Ratnasanry S, Francis P, Handley M, Karp R, Shenker S. A scalable content-addressable network. In: Proc. of the ACM SIGCOMM 2001. San Diego: ACM Press, 2001. 168−175.

[9] Gnutella. http://www.gnutella.com

[10] Lü Q, Cao P, Cohen E, Li K, Shenker S. Search and replication in unstructured peer-to-peer networks. In: Proc. of the ACM SIGMETRICS 2002. ACM Press, 2002. 258−259.

[11] eMule. http://www.emule.net/

[12] BitTorrent. http://www.bittorrent.com/

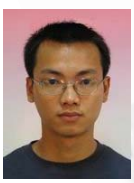
[13] Navigli R, Velardi P, Gangemi A. Ontology learning and its application to automated terminology translation. IEEE Intelligent Systems, 2003,18(1):22−31.

[14] Liu HY. Analysis of resource characteristics and user behavior in P2P file sharing system maze [MS. Thesis]. Beijing: Peking University, 2005 (in Chinese with English abstract).

[15] Has J, Kamber M. Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers, 2000. 225−243.

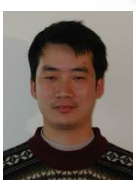
**附中文参考文献**:

[14] 刘翰宇.P2P 文件共享系统 Maze 中资源及用户行为特征分析[硕士学位论文].北京:北京大学,2005.



**邱志欢**(1982－),男,广东汕尾人,硕士生,主要研究领域为对等网络理论、技术.

**代亚非**(1958－),女,教授,博士生导师,CCF 高级会员,主要研究领域为分布式系统,网络存储,P2P 计算.



**肖明忠**(1970－),男,博士,主要研究领域为对等网络,信息检索技术.