

# 基于文件流行度的无结构 P2P 网络搜索机制

吴功宜, 刘 乾, 王 珺, 杨 阳, 徐敬东

(南开大学信息技术科学学院, 天津 300071)

**摘要:** 针对无结构 P2P 文件共享系统对不同流行度的文件使用相同的搜索策略从而导致大量网络资源的浪费问题, 在研究 Gnutella 协议的基础上, 提出一种基于 LogLog 算法的文件流行度判定机制, 将它与 Expanding Ring 算法结合, 给出一种基于流行度的搜索机制。与 Expanding Ring 相比, 该搜索机制在一定程度上减少网络开销和响应时间, 特别是稀缺文件的响应时间。

**关键词:** 无结构 P2P 网络; 洪泛; Expanding Ring 算法; LogLog 算法

## Searching Mechanism Based on File Popularity in Unstructured P2P Networks

WU Gong-yi, LIU Qian, WANG Jun, YANG Yang, XU Jing-dong

(College of Information Technical Science, Nankai University, Tianjin 300071)

**【Abstract】** According to the problem that traditional unstructured P2P networks use the same searching mechanism for the files of different popularity, which consumes a large amount of network resources, this paper proposes a file popularity evaluating mechanism based on LogLog algorithm by researching Gnutella protocol, and combines it with Expanding Ring(ER) algorithm to present a new searching mechanism. Compared with Expanding Ring, this mechanism can reduce network overhead and response time, particularly the response time of the rare files.

**【Key words】** unstructured P2P networks; Flooding; Expanding Ring(ER) algorithm; LogLog algorithm

Gnutella 采用的洪泛搜索具有节点覆盖率高、查询成功率高、健壮性好、响应时间快等优点, 但它不会根据被查询文件的流行度调整查询规模, 导致对流行文件搜索到过多的结果, 间接地造成网络资源的浪费, 而对稀缺文件则搜索到很少的结果甚至查询失败。针对该问题, 学者们提出了多种改进方法, 如 ER(Expanding Ring)<sup>[1]</sup>, Random Walks<sup>[1]</sup>和 Dynamic Querying<sup>[2]</sup>。本文从文件流行度判定的角度对该问题做进一步研究, 在基于闲谈(Gossip 算法)的文件流行度判定方法<sup>[3]</sup>的基础上, 提出了一种更准确的流行度判定机制, 并把它与 ER 机制相结合, 产生了一种基于文件流行度的、自适应的搜索机制。

### 1 改进的文件流行度判定机制

基于闲谈的判定方法利用全局统计信息判定文件的流行度, 该方法有 2 个方面的缺陷: (1)副本数是用与实际副本数最接近的 2 的幂进行估算的, 因此, 估算结果与实际副本数之间有高达 50% 的偏差。(2)尽管信息交换是在超级节点间进行的, 但由于网络的动态性, 一段时间后, 文件流行度估计值会偏离实际情况。

#### 1.1 改进 1: 基于 LogLog 的流行度判定方法

一个改进是利用 LogLog 算法<sup>[4]</sup>提高文件流行度判定的准确性。LogLog 算法用来统计超大数据集中不同元素的数目, 主要用于数据挖掘、数据库查询优化、网络流量分析等领域。

LogLog 算法首先对数据集中的元素用均匀哈希函数进行哈希得到一组随机分布的  $k$  位哈希值( $k > \lg N_{\max}$ )。对于二进制串形式的哈希值  $x$ ,  $\rho(x)$  表示  $x$  中第 1 次出现 0 的位置, 例如,  $\rho(0\cdots) = 0$ ,  $\rho(1110\cdots) = 3$ 。Durand 和 Flajolet 通过实验

验证, 这组哈希值中最大的  $\rho$  值可以认为是  $\lg n$ , 其中,  $n$  为数据集中不同元素的个数。

为了更准确地估计不同元素的个数, LogLog 算法利用哈希值  $x$  的前  $h$  位作为索引, 将哈希值分成  $m$  组, 其中,  $m = 2^h$ 。假设值  $M(j)$  记录第  $j$  组中最大的  $\rho$  值, 则集合中不同元素的个数  $n$  可以估计为

$$n' = \alpha_m m 2^{\frac{1}{m} \sum_{j=1}^m M(j)} \quad (1)$$

其中,  $\alpha_m$  为修正因子。当  $n \rightarrow \infty$  时, 估计值  $n'$  是  $n$  的无偏估计。 $n'/n$  的标准误差(SE)可由方差(Var)得出:

$$SE(n'/n) = \sqrt{\text{Var}(n')}/n \approx 1.30/\sqrt{m} \quad (2)$$

由式(2)可见,  $m$  值越大, 估计结果越准确。

利用 LogLog 算法改进的文件流行度判定方法的具体实现过程分为以下 3 步:

(1)生成交换信息

初始时, 系统中所有节点为自己拥有的每个文件抛  $k$  次硬币( $k > \lg N$ ,  $N$  为 P2P 系统中节点的数目, 假设正面朝上为 1, 反面朝上为 0), 将  $k$  位二进制数的前  $h$  位作为组号, 每个节点需要为每种文件存储  $2^h$  组交换信息。从  $k$  位二进制数的第  $h+1$  开始, 计算连续“1”的个数  $CT$ , 将  $CT$  值存入相应文件的相应组中。 $CT$  的最大值为  $k-h$ , 该值最少使用  $\lceil \lg(k-h) \rceil + 1$  位二进制数就能够表示, 为了减小节点保存的信

**基金项目:** 天津市应用基础研究计划基金资助项目(07JCYBJC14200)

**作者简介:** 吴功宜(1947-), 男, 教授、博士生导师, 主研方向: 计算机网络, 对等计算; 刘 乾、王 珺、杨 阳, 硕士研究生; 徐敬东, 教授、博士生导师

**收稿日期:** 2008-10-20 **E-mail:** liuqian0909@163.com

息量和节点之间的通信开销，每个文件各组交换信息的长度设置为 $\lceil \lg(k-h) \rceil + 1$ 位。

### (2)通过闲谈方式进行信息交换

每个节点通过闲谈方式与其他节点进行信息交换。在任何一轮信息交换中，每个节点把自己保存的  $CT$  值随机发送给一个邻居节点，当一个节点收到其他节点的信息时，它只保存每个文件各组中较大的  $CT$  值。由闲谈算法可知，经过  $\lg N$  轮信息交换后，每个节点能在很大程度上得知每个文件各组中最大的  $CT$  值。

### (3)计算文件流行度

掌握每个文件各组中最大  $CT$  值后，每个节点根据式(1)估算出各个文件的副本数，用副本数除以系统中节点数  $N$  就能得到该文件的流行度。

在信息交换过程中，为了防止由于文件种类太多而导致节点存储的交换信息过多，如果某节点存储的交换信息的数目大于限制  $L$  时，则丢弃那些  $CT$  值较小文件的交换信息，这并不会影响文件流行度判定的准确性。

对于文件实际副本数为 5、抛硬币得到的  $CT$  值为 0,1,2,3,4 的情况，如果采用基于闲谈的判定方法，估算出的副本数为  $2^4/0.77351 \approx 21$ ；而如果采用改进的判定方法，则估算出的副本数为(假设  $h$  取 3)：

$$\alpha_8 \times 8 \times 2^{\frac{1}{8} \sum_{j=1}^8 M(j)} \approx 0.691 \times 8 \times 2^{\frac{1}{8} \times (1+2+3+4)} \approx 13$$

改进方法的判断结果与真实副本数之间仍然有较大的偏差。分析式(1)可以发现，采用 LogLog 算法估算出的副本数会大于等于  $\alpha_m m$ ，因此，对于副本数小于  $\alpha_m m$  的文件，估算结果不准确。可以采用如下方法解决该问题：

交换信息的第  $h+1$  位为 0 的概率为 50%，因此， $\rho=0$  的概率也为 50%。对于副本数为  $m$  的文件，在该文件的  $m$  组交换信息中，多于  $m/2$  组存储 0 的概率为

$$1 - \frac{1}{2^m} (C_m^0 + C_m^1 + \dots + C_m^{\frac{m}{2}-1})$$

当  $m=8$  时，多于  $m/2$  组存储 0 的概率约为 63.7%；对于副本数小于  $m$  的文件，多于  $m/2$  组存储 0 的概率就更大。因此，如果某个文件的  $m$  组交换信息中多于一半存储的是 0，则该文件的副本数极有可能小于  $m$ ，且可用非 0 值个数的 2 倍来估计。文件副本数可按下式估算(假设在  $m$  组交换信息中，有  $x$  组存储的是 0)：

$$n' = \begin{cases} \alpha_m m 2^{\frac{1}{m} \sum_{j=1}^m M(j)} & x < \frac{m}{2} \\ (m-x) \times 2 & x \geq \frac{m}{2} \end{cases} \quad (3)$$

对于上文抛硬币得到  $CT$  值为 0, 1, 2, 3, 4 的情况，如果采用式(3)的判定方法，则估算出的副本数为  $(8-4) \times 2=8$ 。该估算结果与真实副本数 5 比较接近。

改进后系统的开销略有增加。在改进后的系统中，为了估计文件的副本数，每个节点为每个文件存储的信息为(文件名， $2^h$ 组交换信息)，其中，文件名为描述该文件的一组关键词； $2^h$ 组交换信息的长度为  $2^h \times \{\lceil \lg(k-h) \rceil + 1\}$  位，而原方法存储的信息为(文件名，二进制位串)，二进制位串的长度为  $k$  位。假设  $k=32$ 、 $h=3$ ，则改进后的交换信息串长度为 40 位，比原方法仅多出了 1 Byte。在实际应用中， $h$  值的选取是一个关键问题， $h$  值越大，估计结果越准确，但同时系统开销也越大。在模拟实验中，为了权衡系统开销和判定准确性，取  $h=3$ 。

## 1.2 改进 2: 动态更新判定结果

另一个改进是在查询的过程中，根据查询范围和返回结果数更新文件流行度的判定结果，从而减少网络动态性的影响。

如果流行度判定模块把某文件判定为流行文件，但经过查询后发现，使用一个相对较大的  $TTL$ (如  $TTL$  为 7)还没有得到满意的结果，则可以推断先前对该文件的流行度判定有错，可以将该文件的副本数更改为查询的返回结果数，从而降低该文件的流行度。同样，如果流行度判定模块把某文件判定为稀缺文件，但经过查询后发现，仅仅需要一个相对较小的  $TTL$ (如  $TTL$  为 4)就能返回满意的结果，则可以根据  $TTL$  值和返回的结果数相应调整文件的副本数。在 P2P 系统中，不同  $TTL$  对应不同查询范围，在本文的模拟实验中，不同  $TTL$  对应的查询范围如表 1 所示。如流行度判定模块把某文件判定为稀缺文件，但经过  $TTL$  为 4 的查询后返回了 10 个结果，则可以估算该文件的副本数为  $10 \div 10\% = 100$ 。

表 1 不同  $TTL$  对应的查询范围 (%)

| $TTL=2$ | $TTL=3$ | $TTL=4$ | $TTL=5$ | $TTL=6$ | $TTL=7$ |
|---------|---------|---------|---------|---------|---------|
| 0.2     | 1.5     | 10.0    | 43.0    | 90.0    | 99.5    |

该方法可以在一定程度上适应由节点加入、退出和文件删除、下载而引起的文件流行度偏离网络的实际情况。

## 2 基于文件流行度的 ER 搜索机制

用 ER 搜索机制查询稀缺文件时，通常  $TTL$  达到最大值时都不能返回满意的结果，前几轮  $TTL$  相对较小的查询只会浪费网络带宽、增加响应时间，因此，对于稀缺文件，如果采用相对较大的初始  $TTL$  进行搜索，则可以同时节省查询流量、减少响应时间。

本文提出将文件流行度判定与 ER 搜索机制相结合的方法，其基本思想是根据文件的流行度设置不同的初始  $TTL$  值：对于流行文件，采用相对较小的初始  $TTL$ ，对于稀缺文件，采用相对较大的初始  $TTL$ 。

基于流行度的 ER 搜索机制由 3 个模块组成：文件流行度判定模块，参数选择模块和搜索模块。流行度判定模块采用改进的闲谈算法判定文件的流行度；参数判定模块负责根据流行度设置不同的初始  $TTL$ ；搜索模块进行 ER 搜索，并将查询结果及相应的  $TTL$  值反馈给流行度判定模块，用于更新流行度判定结果。

如何根据文件流行度选取合适的初始  $TTL$  是提高查询效率的关键，可以采用如下方法：每个节点保存一张文件流行度与初始  $TTL$  对应的关系表，当节点需要查询流行度为  $p$  的文件时，首先查找关系表，得到与流行度  $p$  对应的初始  $TTL$ ，然后设置初始  $TTL$  并进行搜索。表 2 是关系表的一个示例。因为对应关系表只要建立一次就能长期使用，所以该表可以通过运行原始的 ER 搜索机制获得：对不同流行度的文件利用原始的 ER 搜索，记录它们获得满意结果时的  $TTL$  值，然后根据此结果反向推断不同  $TTL$  所对应的流行度区间。

表 2 流行度与初始  $TTL$  对应关系的示例

| 流行度                                      | 初始 $TTL$ |
|--|----------|
| $[20 \times 10^{-3}, 1)$                 | 3        |
| $[4 \times 10^{-3}, 20 \times 10^{-3})$  | 4        |
| $[1 \times 10^{-3}, 4 \times 10^{-3})$   | 5        |
| $[0.5 \times 10^{-3}, 1 \times 10^{-3})$ | 6        |
| $[0, 0.5 \times 10^{-3}]$                | 7        |

### 3 模拟实验

在实验中用 BRITE 生成覆盖网络拓扑图并周期性地让节点加入或离开网络,从而模拟总数为 40 000 并始终有 25 000 个节点存活的覆盖网络。该覆盖网络节点的平均度数为 6。并使用了 2 000 个不同的文件,以 Zipf 分布模拟查询的分布和文件副本数的分布<sup>[5]</sup>:流行度最高的 10%的文件占据了文件副本总数的 50%,并收到了半数以上的请求。在设置中,最流行的文件副本数达到 600 个,存储在约 2.4%的节点上,而最不流行的文件只在 0.008%的节点上存有拷贝。

#### 3.1 文件流行度的判定结果

系统参数设为  $k=24$ 、 $h=3$ 。图 1 是在系统运行初期没有节点加入、退出和文件删除、下载的情况下得到的文件流行度判定结果。可以看出,改进方法估算的文件副本数比原方法更接近真实副本数。由于对副本数较少的文件采用了特别的估算方法,因此改进方法对副本数较少的文件判定得非常准确。尽管在查询过程中会根据查询结果更新流行度判定结果,但由于网络的动态性,经过一段时间的查询后,流行度判定结果还是在一定程度上偏离了网络的实际情况,见图 2。

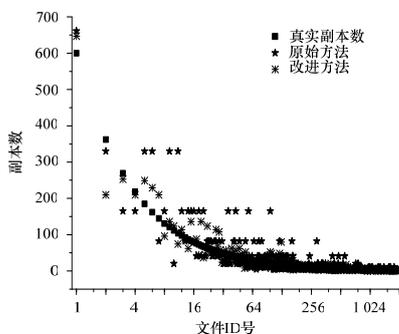


图 1 文件流行度的判定结果

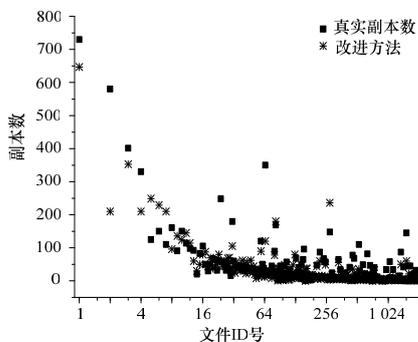


图 2 一段时间后文件流行度的判定结果

#### 3.2 基于流行度的ER搜索机制的测试结果

另外分别模拟了 ER 和基于流行度的 ER(假设命中结果数大于 8 时满意)。模拟 ER 时,初始  $TTL$  设置为 3,当没有返回满意结果时, $TTL$  值加 1 后再次查询,直到返回满意结果或达到最大  $TTL$ ,搜索过程结束。模拟基于流行度的 ER 时,根据文件流行度设置相应的初始  $TTL$  值,当没有返回满意结果时, $TTL$  值加 1 后再次查询,直到返回满意结果或达到最大  $TTL$ ,搜索过程结束。

图 3 和图 4 给出了算法查询平均消息数和平均响应时间的比较。从图 3 可以看出,基于流行度 ER 比 ER 的平均消息数有明显的降低,当最大  $TTL$  为 7 时,基于流行度 ER 比 ER 的平均消息数降低了 30.1%。从图 4 可以看出,基于流行度 ER 的平均响应时间比 ER 减少了很多,几乎接近洪泛。

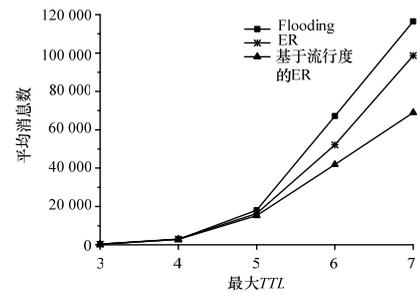


图 3 不同最大  $TTL$  值对应查询的平均消息数

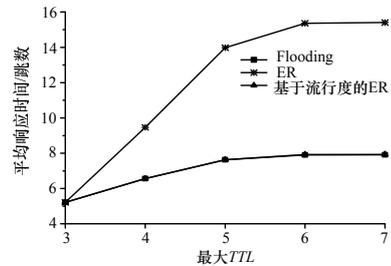


图 4 不同最大  $TTL$  值对应查询的平均响应时间

为了比较查询稀缺文件的响应时间,实验中模拟查询了副本数小于 8 的所有文件。如图 5 所示,与 ER 相比,基于流行度 ER 的平均响应时间减少了很多,当最大  $TTL$  为 7 时,平均响应时间减少了约 60%。

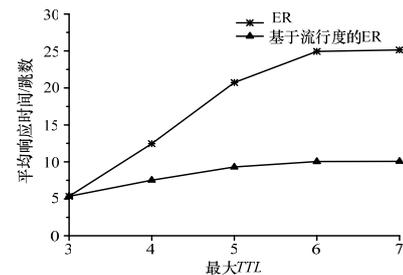


图 5 不同最大  $TTL$  值对应查询的平均响应时间

此外,这 3 种搜索机制在设置相同最大  $TTL$  值的情况下能获得相同的查询成功率和满意度。

### 4 结束语

本文提出一种基于文件流行度的 ER 搜索机制。模拟实验显示,与原 ER 搜索机制相比,新机制在响应时间和资源消耗等方面都有显著的提高。今后将继续研究提高文件流行度判定准确性的方法,使之更加适应 P2P 网络的动态性,并把流行度判定与其他搜索机制相结合,进一步提高查询效率。

#### 参考文献

- [1] Lv Qin, Cao Pei. Search and Replication in Unstructured P2P Networks[C]//Proc. of ICS'02. [S. l.]: ACM Press, 2002.
- [2] Fisk A. Gnutella Dynamic Query Protocol v0.1[Z]. [2008-02-11]. [http://www.limewire.com/developer/dynamic\\_query.html](http://www.limewire.com/developer/dynamic_query.html).
- [3] Zaharia M, Keshav S. Gossip-based Search Selection in Hybrid P2P Networks[C]//Proc. of IPTPS'06. [S. l.]: ACM Press, 2006.
- [4] Durand M, Flajolet P. LogLog Counting of Large Cardinalities[C]//Proc. of ALGO'03. [S. l.]: ACM Press, 2003.
- [5] Chu J, Labonte K, Levine B N, et al. Availability and Locality Measurements of P2P File Systems[C]//Proc. of SPIE'02. [S. l.]: ACM Press, 2002.

编辑 张帆