

一种高效的无结构对等网络搜索机制^①

An Efficient Search Algorithm for Unstructured Peer-to-Peer Networks

徐 浩 欧阳松 (中南大学 计算机科学与技术系 湖南 长沙 410083)

摘 要: 点对点网络已发展成为当今最为流行的资源共享系统之一。但是目前的搜索算法的效率通常很低,并且一般只遵循很少的规则,因此往往会给网络带来大量的冗余信息。通过对冗余消息产生的本质原因进行分析,提出了预搜索优化算法。该算法通过在预搜索阶段所获取的实时网络拓扑结构和资源的信息,制定有效的优化设置,从而在正式搜索时控制冗余消息的产生,提高资源搜索的效率。

关键词: 无结构 P2P 网络 预搜索 环结构 忽略机制

1 引言

P2P 系统发展至今,已经演化出三种主要的体系结构:集中式,结构化分布式和非结构化分布式。非结构化分布式 P2P 系统资源的查找通过相邻节点之间以泛洪方式转发查询请求来实现,从而能够较快地地发现目的资源结点,面对网络的动态变化体现了较好的容错能力,因此具有较好的可用性。同时可以支持复杂查询,如带有规则表达式的多关键词查询,模糊查询等。以 Gnutella 协议生成的网络就是典型的非结构化分布式结构。可是,随着网络节点的不断增多,网络规模不断扩大,通过泛洪方式定位资源节点将造成网络流量急剧增加,由此产生出大量的冗余数据。这些冗余数据必然会导致网络的负担过重,部分低带宽节点甚至会因网络资源的过载而失效。并且,搜索效率大大降低。针对泛洪转发的不足,一些改进算法被提出,如本地索引(local index)^[1]算法,迭代加深(iterative-deepening)算法,以及 k 遍历随机游走(k-random-walker)^[2]算法等。

本文主要讨论非结构化分布式 P2P 网络。针对大规模 P2P 网络的信息冗余量问题,提出预搜索机制。该机制要求在节点搜索资源之前进行深度较低的小规模的预搜索,然后根据当前实时的拓扑结构进行优化处理和设置,进而再正式搜索时减少信息冗余量,提高搜索的效率。

2 问题描述

随着非结构化分布式网络的规模不断扩大,传统的资源搜索算法所采取的泛洪转发方式会产生大量的查询数据流量,很容易形成网络流量阻塞。

首先,消息的向前转发不能辨别出被请求的节点是否已经收到同样的请求消息。目前,无结构分布式 P2P 系统通常的处理方式是被请求节点忽略掉重复的请求消息,而无法探测并阻止重复消息的转发。图 1 所示是一个局部网络拓扑图,它一共包含 6 个对等节点。假设请求消息从节点 B 发出,目标节点会忽略掉重复收到的相同消息,消息经每条路由转发的时间开销大致相同,并且节点 C 收到节点 A 的消息的时间略早于节点 E。经过若干次转发后,其他五个节点会先后收到消息请求。我们可以看到,在全部 11 次消息中,存在大量的重复消息。其中节点 C 和节点 F 会重复收到 3 次相同的信息,节点 D 和节点 E 重复收到 2 次相同的信息。因此,总共产生了 6 条冗余消息,占消息总数的 54.5%。

分析图 1,我们可以发现,网络中的环结构是造成重复请求消息的根本原因。而实证结果表明,大部分大规模真实网络中的节点倾向于聚集在一起,存在小世界现象^[3]。因此,环结构在对等网络中必然大量的存在。

① 收稿时间:2008-12-18

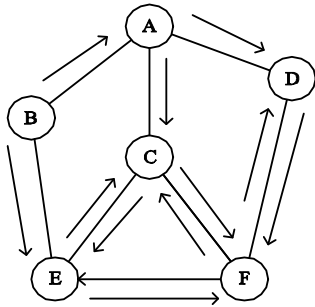


图 1 局部网络消息转发模拟图

3 预搜索优化机制

通常，分布式 P2P 网络的每个对等节点维护着一张邻居节点列表，在节点连入网络之初通过 p2p 网络中央索引器获取得到。之后列表会随着邻居节点的加入或断开而不断更新。优化机制为了维护预搜索获取并分析得出的优化信息，在原来邻居节点表的基础上，增加优化转发信息表项：忽略节点源(Ignore From)和控制忽略的时间戳(Timestamps)。

每当节点发起预搜索，它向所有相连的节点(邻居节点)发送请求消息，我们定义此时的 $TTL = TTL_{thresh}$ ，由于此时预搜索的目的是在小区域内获取实时网络拓扑结构，因此 TTL_{thresh} 值会根据实际节点的能力设置一个较小的值。当源节点发出的消息请求后，它的所有跳数(Hop)小于 TTL_{thresh} 的近程节点都将获得请求消息。根据请求消息到达的时间和顺序，各个收到消息请求的目的节点进行分析处理，并生成各自邻居节点列表的相关优化转发表项。

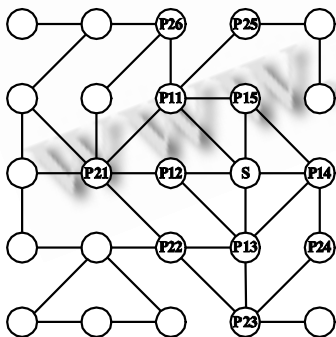


图 2 局部网络拓扑结构模拟图

图 2 表示 P2P 覆盖网拓扑结构的一个示例，圆圈表示对等节点，实线表示 P2P 节点与邻居节点间的连接。其中，S 节点表示首次预搜索的源节点，P11-P15

节点是 S 节点的初始相连(邻居)节点。假设此时源节点定义 $TTL_{thresh} = 2$ ，则网络中所有的 P 节点就会收到 S 节点在预搜索阶段发出的请求消息。

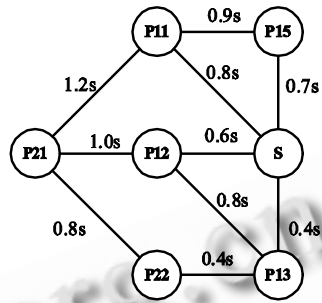


图 3 消息转发模拟图

图 3 模拟了一次的 S 节点预搜索的消息转发的局部情况，两节点间实线附近的数值表示通过该连接执行消息转发所要开销的总时间。我们可以看到，节点 P12 此时会收到源节点 S 产生并发出的同一个消息请求两次，第一次来自源节点 S，它处理请求，并进一步向前转发给 P21 和 P13；第二次的请求消息来自 P13，在距离第一次收到请求消息的 0.6s 后。此时，P12 会忽略掉这个请求重复的消息。而在忽略的同时，预搜索优化机制生成优化转发信息：将发出重复消息的邻居节点 P13 设置为自己的忽略节点，并且当转发消息的途经列表包含了 S 节点时忽略机制生效。实际做法是将 P12 节点的邻居列表中的 P13 节点对应的 Ignore From 项插入 S 节点，如表 1。从而表明当某一转发消息的若曾经经过 S 节点，那么该消息不会被节点 P12 向前转发给节点 P13。时间戳 Timestamp 项控制忽略节点的失效时间。失效时间一旦到达，清除相关 IGNORE From 项，对应的邻居节点取消忽略。

表 1 节点 P12 的邻居节点列表

Peer ID	Peer IP	IGNORE From	Timestamp
P13	***.*	S	
P21	***.*	P22	
S	***.*	P13	

如果此时节点 P22 在预搜索阶段发出另一个请求消息，P12 会分别从 P13 和 P21 收到两次该请求消息。于是，它根据后到的消息将相关节点加入忽略列表。这次，它会忽略掉 P21，因为从 P13 过来的 P22 请求消

息用了 1.2s, 而从 P21 过来的同样消息在收到 P13 的消息后的 0.6s 才被 P12 收到。同理, 当节点 P13 进行首次预搜索后, P12 的邻居节点列表中, 邻居节点 S 也会当消息经过 P13 时被 P12 忽略掉。

我们可以看到, 预搜索机制并不是为自身的邻居节点列表生成优化转发信息, 而是为自己的近程节点生成优化信息。近程节点是在预搜索阶段通过 TTL 的设置确定的, 即在预搜索阶段收到源节点消息的节点为源节点的近程节点。这些近程节点在预搜索阶段所生成的忽略机制, 为源节点在正式搜索阶段提供优化转发信息, 提高搜索效率, 减少数据冗余的产生。由于对等网络的本质, 各个对等节点本身既是源节点, 又会是其它节点的近程节点, 因此, 忽略机制对于每一条请求消息, 会在整个转发过程中有效。而不仅仅局限于该消息请求源节点的近程节点范围内。

我们用预搜索优化机制来处理前文中图 1 所示的情况, 假设此时节点 B 定义的 TTL_{thresh} 值等于 3, 通过前期预搜索的忽略机制处理后的路由路径, 在覆盖了全部节点的同时, 完全抑制了冗余信息的产生, 正式搜索时的消息转发情况如图 4 所示。

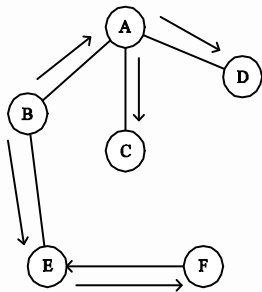


图 4 消息转发模拟图

我们考虑图 5 的情况, 4 个节点相互连接形成一个矩形结构, 假设节点 A 与节点 D 之间的连接由于某种原因形成网络拥塞, 此时连接的转发开销为 0.8s, 其他连接的转发开销为 0.2s。

设置 $TTL_{thresh}=3$, 分别从 ABCD 四个节点发出预搜索请求。根据预搜索忽略机制, 从节点 A 发出的请求消息最终会终止在节点 D。节点 D 收到来自节点 A 和节点 C 的两次对同一资源相同的重复请求。由于通过路由 AD 转发的消息迟于另一个消息, 因此 D 将节点 A 设为忽略节点, 当请求消息途经节点表中包含了节点 A 时, 它会忽略掉它的邻居节点 A。另一方面,

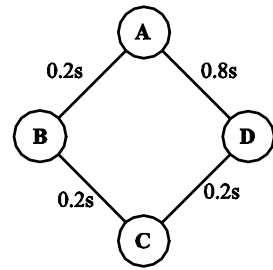


图 5 网络拥塞模拟图

当请求消息从 B 中发出时, 节点 A 会被添加到节点 D 的忽略邻居列表中, 因为节点 D 从节点 C 收到的消息比从节点 A 收到的消息要早 0.6s; 而节点 D 也会被设置为节点 A 的忽略邻居节点, 因为节点 A 从节点 D 收到的消息要比从节点 B 收到的消息晚 1.0s。同样, 我们可以分析出从节点 C 和节点 D 的预搜索发出请求消息后, 优化忽略机制根据消息到达的先后次序所处理的优化结果。最终, 任何途经四个节点之一的请求消息, 在到达节点 A 或者 D 后, 拥塞路由 AD 或者 DA 就会被暂时地忽略掉, 消息转发将不经过这条时间开销较大的拥塞路由。因此, 在预搜索优化机制下, 网络拥塞问题可以得到有效地控制。

4 仿真实验

研究证实, 无结构分布式对等网络的典型代表——Gnutella 网络, 满足幂率分布。因此在仿真实验中, 我们利用 BA 系统模型^[4]所具有的幂律分布特性, 以它为原型构建仿真系统。BA 模型之所以能够描述具有幂律分布特征的无标度网络(scale-free network), 归结于它的两个关键特性: 增长特性和优先连接特性。增长特性描述了网络系统是不断扩大的; 优先连接特性则说明新加入的节点更倾向于和那些具有较大连接度的节点连接。这也与现实中资源分享型对等网络的特点相符合。BA 无标度网络模型的构造算法如下:

(1) 增长: 从一个具有 m_0 个节点的网络开始, 每次引入一个新的节点, 并且连到 m 个已存在的节点上, $m \leq m_0$ 。

(2) 优先连接: 一个新节点与一个已经存在的节点 i 相连接的概率 Π_i 与节点 i 的度 K_i , 节点 j 的度 K_j 之间满足下式关系:

$$\Pi_i = \frac{k_i}{\sum_j k_j}$$

上式中分母求和为取遍网络已有的各节点的连接。在 t 时间步后，算法生成的网络有 $N=t+m_0$ 个节点和 mt 条边。

仿真系统是在 Linux 平台上用 C++ 搭建，它由 N 个节点构成，每个节点从一个容量为 M 的 key 池中随机选择 m 个 key 来代表本地文件特征。表 2 是我们基于 BA 模型的实验参数设定。

表 2 仿真试验参数设置

Parameter	m_0	m	t	Key 池 M	TTL_{thresh}
Value	10	3	200	200	2

我们定义 $\sum_j I_i$ 表示当消息转发经过节点 i 时，其近程节点 j 的忽略列表。例如： $\sum_5 I_3=(1,4,8,17)=4$ 表示当节点 5 收到来自节点 3 (即消息转发路径包含节点 3) 的请求消息时，它将忽略它的 4 个邻居节点 (1、4、8、17)，即消息不会再向前转发给这四个邻居节点。

$\sum_i \sum_j I_i$ 表示对等网络全部节点预搜索后所生成的忽略节点总数， $\sum_i \sum_j I_i / N$ 表示对于单个源节点，其全部近程节点生成忽略节点总数的平均值。

我们随机选择源节点，并给定一随机 key 值，让源节点搜索存储该 key 值得目标节点。表 3 是仿真试验根据上述设置所产生的一组实验数据：

表 3 $TTL_{thresh}=2$ 时所生成的仿真试验数据

	网络平均度 D	平均近程节点数 J	平均近程节点忽略节点总数 $\sum_i \sum_j I_i / N$
一	5.91	80.88	69.76
二	5.96	83.56	76.41
三	6.13	67.39	75.71
四	5.59	83.98	72.05
五	5.36	79.33	61.75
平均	5.79	79.03	71.14

由上表可以看出，由于仿真网络满足幂率分布，存在一定数量的高聚集度节点，因此在平均度约等于 6，值等于 2 的情况下，邻居节点数大大高于 36 的理

论值：平均每个节点有 79 个近程节点。而这些近程节点生成的忽略节点总数约为 71 个，平均每个近程节点会有 0.9 个邻居节点被忽略。

同时，消息转发是向前传递的，当消息转发路径新增加一个节点时，针对此节点的忽略机制将会叠加上去。当 $=2$ 时，近程节点的跳数小于等于 2，忽略机制对消息转发的作用效果将会有 2 次叠加。因此预搜索设置的值越大，一方面可以发现由更多的节点所构成的环结构，从而分析出更多的冗余消息；另一方面，优化机制所具有的忽略迭加特征所产生的功效越大，对冗余消息的控制越为高效。

5 结论和展望

本文针对无结构分布式 P2P 网络的搜索算法进行研究，提出了预搜索忽略优化算法。它要求节点在进行资源搜索之前先小规模预搜索，从而获得局部网络的拓扑结构信息。根据这些掌握的信息，我们可以通过加入忽略机制来优化网络拓扑结构，逻辑上拆分环结构，而这正是产生冗余消息的主要原因之一。

另一方面，TTL 设置的不合理会直接导致泛洪式搜索效率的低下，而预搜索本身也一次资源定位的过程，因此当预搜索的被请求节点数达到一定数量时，我们可以根据局部区域内的资源数来预估整个网络的资源密度。这样，就可以为正式搜索的 TTL 值设置提高了有实际意义的参考。从而在另一方面提高搜索的效率，减少冗余消息的产生。这正是我们接下来要进行的工作之一。

参考文献

- 1 Yang B, Garcia-Molina H. Improving search in peer-to-peer networks. IEEE, 22nd ICDCS, Vienna, Austria, 2002:2-14.
- 2 Lv Q, Cao P, Cohen E, Li K, Shenker S. Search and replication in unstructured peer-to-peer networks. Proc. of the 16th ACM Int'l Conf. on Supercomputing (ICS 2002). New York: ACM Press, 2002: 84-95
- 3 Watts DJ, Strogatz SH. Collective dynamics of small-world networks. Nature, 1998,393:440-442.
- 4 Albert R, Barabási L. Statistical mechanics of complex networks. Reviews of Modern Physics, 2002,74:47-97.