

Peer-to-Peer 网络中文件查询的研究^{*})

王 涛 卢显良 侯孟书

(电子科技大学计算机学院 成都 610054)

摘 要 随着 P2P 网络应用研究的热化,文件查询即给出文件名后如何确定文件的存放位置,已经成为大规模分布式网络中最常遇到的问题之一。文件查询的效率直接影响到整个 P2P 网络的性能。本文从 P2P 网络的基本现状出发,给出了目前文件查询在非结构化,结构化和松散结构化 P2P 网络中的解决方案并进行了深入分析和比较。文章最后指出了目前 P2P 网络中文件查询算法的缺点和今后的研究方向。

关键词 P2P, 文件查询, 分布式

Research on the Search Problem in Peer-to-Peer Network

WANG Tao LU Xian-Liang HOU Meng-Shu

(College of Computer Science and Engineering, UESTC, Chengdu 610054)

Abstract With the development of P2P network, the search problem is coming into been one of the most critical problems in distributed systems. The efficiency of search is the key of the whole P2P network. In this paper, the present situation of P2P network is described. The solutions of search problem in unstructured, structured and loosely structured P2P systems are provided. After analyzing and comparing, we propose shortcomings of the present search and give some suggestions for the future research on P2P systems.

Keywords P2P, Search, Distributed

1 引言

近年来,一种名为“Peer-to-Peer”(P2P)的新型网络结构大行其道。这种网络不通过中心服务器,而在计算机之间进行直接访问。“通过直接交换来共享计算机资源”是 P2P 网络的鲜明特征。

Peer-to-Peer 的研究热潮正刮遍整个互联网。有大量的研究团体和公司开发出了实验或实用的 P2P 系统,这些系统最主要的目的是通过互联网连接孤立的计算机用户,实现文件的存储、共享、交换。其中比较著名的有 Napster, Gnutella^[1], Kazaa, Morpheus, Freenet^[2], Chord^[3,9], PAST^[4], CAN^[5], Ocean Store, BitTorrent, Free Haven 等。

“文件查询”是指在网络中如何根据文件名确定文件的存放位置。在大规模分布式网络中它是最常遇到的问题之一。

一个节点如何在 P2P 网络中定位自己需要的文件,是能够直接知道哪个节点保存了自己需要的文件,还是采用一种随机搜索(random search)的方式在网络中查找文件?不同的网络结构有不同的解决方案。根据网络结构的不同可以将 P2P 文件存储系统分为三类:

非结构化的(unstructured):在这种网络结构中,文件的存放位置与 overlay 网络拓扑结构完全无关。节点在存取一个文件的时候,不知道文件的具体存放位置,只能通过一种“洪泛”(Flooding)的机制发起随机的搜索请求,查询大量的节点,询问它们是否有符合查询条件的文件。这种系统的好处在于节点的数量对系统来说完全透明,节点的加入退出相对自由。但是要想获得较高的文件查获率,必须在系统中尽

可能地散布查询消息,当系统规模很大的时候,将使得通讯开销急剧增加。所以这种系统一般被认为不具有可扩展性。代表系统是早期版本的 Gnutella。

结构化的(structured):在这种网络结构中,文件 ID 和文件存放位置之间存在一个映射关系,文件的存放位置可以被精确定位。结构化的 P2P 网络不“直接”利用 IP 层的路由算法,而是构建自己的 overlay 网络路由算法。当节点发出对一个文件的查询请求时,这个请求可以通过转发,路由到存放文件的节点处。在结构化的网络结构中,只要知道文件的 ID,就可以沿一定的路径找到文件,避免了随机搜索的随意性。结构化网络的缺点在于较难适应高动态的系统环境。如果节点加入和离开的随意性太大,将导致节点上的路由表维护困难。代表系统有 Chord、CAN、PAST、Tapestry 等。

松散结构化的(loosely structured):这种网络结构介于结构化和非结构化之间。文件的位置是由文件存放时沿途留下的“线索”(hint)来指引。节点根据线索,一步一步尝试找到文件。由于文件定位采用的是“尽量”(do best)策略,因此有可能找不到系统中存在的文件。代表系统是 Freenet。

2 非结构化系统中的文件查询

最简单的方法是在 P2P 网络中维护一个中心名字服务器,当节点在加入系统时,向中心服务器报告自己机器上的共享资源,中心服务器编制记录项。当一个客户想查找某个文件的时候,向中心服务器发出文件定位请求,中心服务器根据自己掌握的数据,查出文件所在的节点位置,返回给客户。如图 1 所示, Napster 采用的就是这种方法来实现文件查询的。

^{*}) 电子信息产业发展基金资助项目,编号:信运部[2002]-546。王 涛 博士研究生,主要研究方向:分布式文件系统, P2P 计算。卢显良 教授,博士生导师,主要研究方向:计算机网络,操作系统,信息安全。侯孟书 博士研究生,主要研究方向:分布式文件系统,分布式存储。

Napster 是第一个成功实现商业化的 P2P 软件,到目前为止,已经拥有了数千万的用户,但正如集中式分布式系统的通病,这种文件查询存在单点故障。

与 Napster 不同,Gnutella 中取消了以中央服务器为核心的目录式结构,而是采用 Flooding 算法按广度优先原则搜索文件位置。为了查找某个文件,节点首先向与自己相邻的所有活动节点以广播的方式发送一个查询请求包。其它节点接收到查询请求后,检查本地是否有符合条件的文件,如果有,则按查询请求包的路径返回一个响应包,响应包中包括响应节点的 IP 地址、服务端口等信息;如果没有,则节点再将该请求向自己的邻居转发,直到查询包的 TTL(Time To Live)计数器减为 0 为止。如果发起查询的节点成功地收到返回的响应包,则直接与目标节点建立连接。图 2 显示了一个典型的 Gnutella 查询过程。

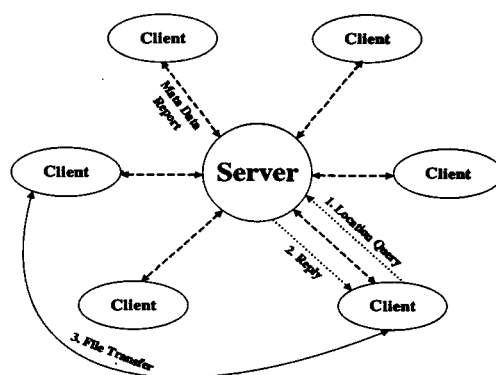


图 1 Napster 文件查询过程

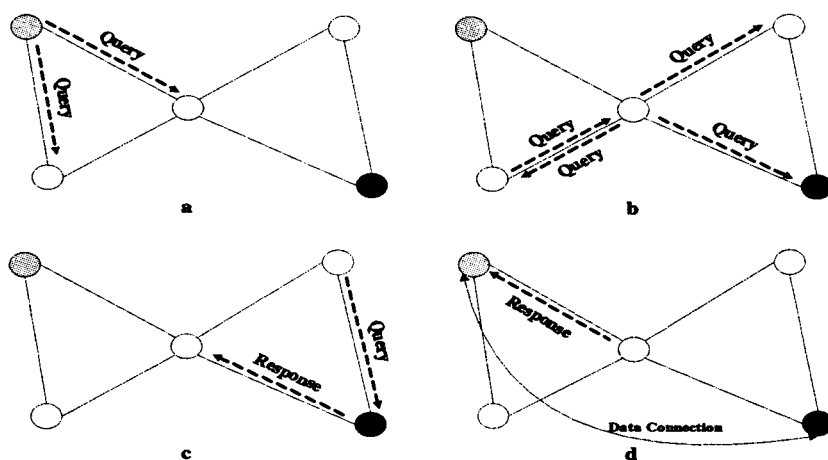


图 2 Gnutella 文件查询过程

上述 P2P 系统采用了均衡查找的算法。这些系统中所有的节点在文件查找时都具有平等的地位,每个节点都包含一小部分的查询路径。这种类似广播的文件查询算法收敛性不是很好,广播消耗了大量的带宽,而且容易形成回路。

增加“超节点”可以解决上述的问题,这些节点保存着自己所在区域内全部节点的文件索引。系统中的“超节点”并不固定,一般通过选举算法动态指定。代表系统包括 Kazza、Morpheus。当然“超节点”的失效也会影响到整个 P2P 网络的使用。

3 松散结构化系统中的文件查询

Freenet 采用了一种改进的均衡查询策略。查询请求从一个节点转发到另一个节点,直到在非结构的路由表中发现了所要的文件。

与 Gnutella 广度优先的“Flooding”搜索机制不同,Freenet 系统采用了一种深度优先的“链式”(chain-mode)文件查询机制。Freenet 中的每个文件都有一个全局唯一的文件标识(File ID),系统中的每个节点维护着一张动态路由表,记录了系统中部分节点的地址和这些节点上可能保存的文件。当节点收到一个文件查找请求的时候,先查看本地是否有这个文件,如果没有,则从本地路由表里选取一个邻居节点,这个节点拥有的文件 ID 是路由表中最靠近所查询 ID 的,并将查询请求转发给那个节点;如果本地有所查询的文件,节点将文件数据返回给向自己查询的节点。这样,文件数据将

沿查询消息的来路返回给最初请求文件的节点,并沿途留下副本。查询请求带有一个 TTL 计数器,这使得一个查询请求有可能找不到所需文件,是一种“尽力”(do-best)的查询方式。中间节点在返回数据的时候都会宣称自己是数据源,以保持文件属主信息和存放位置的“匿名性”,这也是 Freenet 区别于其他系统的最大特点。图 3 是一个典型的 Freenet 文件定位过程。

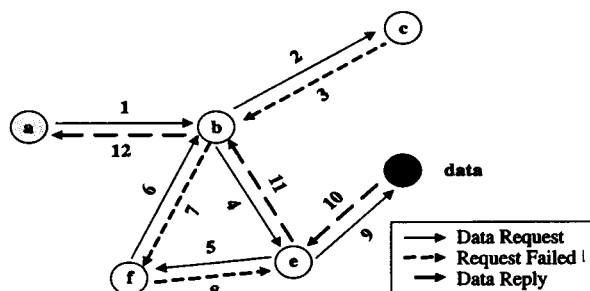


图 3 Freenet 文件定位过程

4 结构化系统中的文件查询

在基于分布式散列表^[6](Distributed Hash Table, DHT)的 P2P 网络中,文件和节点根据分布式散列算法,分别被赋予一个全局唯一的标识符,文件根据自己的标识符,存放在相应的系统节点中。这些节点分别保存了一部分的路由信息,

从而可以被方便高效地组织在一起,形成一个 overlay 网络。在 DHT 网络结构中,系统通过节点间的消息转发来实现文件的定位操作。

DHT 网络中文件查询的主要过程是:给要共享的文件一个唯一的文件名,系统将用一些算法(如 SHA-1 算法^[7])将文件名映射成一个数字标志符 *key*,然后交给 *lookup(key)* 函数,*lookup(key)* 函数可以找到标志符 *key* 所代表文件的具体存放位置,并将结果返回。

4.1 一维空间上的文件查找

结构化 P2P 网络的特点是文件 ID 和存放位置之间存在一个映射关系,文件的存放位置可以被精确定位。Chord 是一种最简单的结构化 P2P 文件存储系统。在 Chord 中,每个文件和节点都有一个全局唯一的 ID。每个节点保存一张指向所有活动节点的“finger table”,ID 为 *n* 的节点的“finger table”表中第 *i* 项指向 ID 空间中最小最接近 $n + 2^{i-1}$ 的节点。这样,所有节点被 ID 空间串连成了一个逻辑上的环。ID 为 *k* 的文件存放在 ID 与 *k* 相同或者最近的节点上。当某个客户查找文件 *k* 时,查询请求沿逻辑环跳跃并传递到存放文件的节点。图 4 是 Chord 系统中一个节点 N8 的 finger table 示意图。

PAST 中 Pastry 路由算法为系统中每个文件分配全局唯一的 160 位的文件标识(fileID)。系统中的每个节点也有一个 128 位的节点标识(nodeID),通过这个标识,系统中的节点被组织在一个环形名字空间中,这个名字空间的范围是 $0 \sim 2^{128} - 1$ 。fileID 中的前 128 位用来指定存放文件的节点;文件的 *k* 个副本被存放在 *k* 个 nodeID 与 fileID 前 128 位最接近的节点上。这样,只要知道一个文件的 fileID,就能找到存放它的节点。由于 fileID 和 nodeID 都是通过安全散列函数(SHA-1)产生,这保证了 nodeID 与节点地理位置无关,并使得文

件能够尽可能平均地分配到系统的所有节点中,实现节点间存储负载的相对平衡,避免多数文件集中到少数节点,造成节点负载过重。

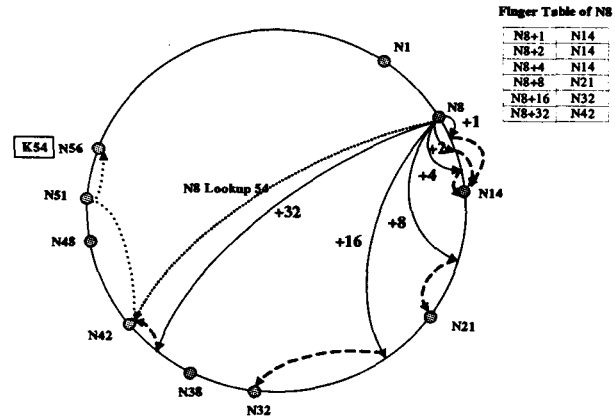
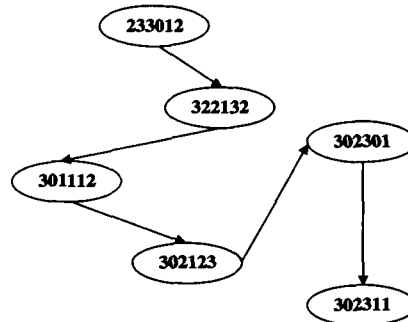


图 4 Chord 系统 finger table 示意图

Pastry 中文件查找过程如下:系统中的 nodeID 和 fileID 被按照 2^b 进制划分。每个节点维护一张 $128/2^b$ 行、 2^b 列的路由表。路由表中第 *r* 行存放着与本节点的 nodeID 前 *r* 位数字相同但与第 *r*+1 位数有着不同值的 2^b 个节点对应的 IP 地址。当一条查询文件的消息在 Pastry 中的节点间传递时,在路由表中选择一个 nodeID 比当前节点多一位与 fileID 相同的节点,并将消息转发给它。这样,每经过一次路由,消息就向存有 fileID 文件的节点更靠近了一步,最终到达该节点。通过使用这种路由方式,如果系统中共有 *N* 个节点,一个文件查询消息可以在 $\log_2 N$ 步之内到达目标节点。图 5 显示了 nodeID 为 233012, *b*=2 的节点的路由表示例,右图是从 233012 到 302311 所经过路由节点。

023462	132142	322132
203213	214313	222113
230132	231322	232310
233110	233203	233312
233003	233112	233301
233010	233011	233013

a. Route table of nodeID 233012



b. Route from 233012 to 302311

图 5 Pastry 节点路由表和路由过程

与 Pastry 类似的路由算法还有用于 OceanStore 的 Tapestry^[8],它们都是结构化的 P2P 路由算法。

4.2 多维空间的文件查找

Can 使用一个 *d* 维笛卡儿坐标空间来实现 DHT,坐标空间被划分成一个一个的“区”。系统中每个节点隶属于一个“区”,并且被这个“区”的边界所独立标识。

系统中每个节点保存有一张包含全部邻居节点的路由表。文件查询是靠转发查询消息来实现的,转发沿着近似直线的路径从请求者到保存数据的节点。在收到一个查询请求后,节点将它转发给坐标空间中最相邻的并且保存有该文件的节点,然后主动停止查询转发。

当新的节点加入到 Can 中时,它首先在坐标空间内选择

一个随机的点 *P*,并且向网络中已经存在的节点发出请求,来找到包含点 *P* 的“区”和该“区”现有节点 *n*。节点 *n* 将自己的“区”划分为两个,将其中一个赋予新的节点。新的节点可以初始化自己的路由表,因为除了 *n* 节点外,新节点所有的邻居,都是 *n* 节点的邻居。一旦加入后,新节点会通知它的邻居节点,让它们更新各自的路由表。

当一个节点离开时,它会将自己的“区”送给邻居节点。如果合并两个“区”可以产生一个合法的“区”时,那这两个“区”就会合并成一个较大的“区”了;如果不行,邻居节点就将临时掌管两个“区”。为了处理节点失效的情况,Can 允许失效节点的邻居取代失效节点来掌管“区”。这样会带来一个潜

(下转第 145 页)

表2 神经网络仿真的迭代次数和总误差

隐节点数	迭代次数	总误差	隐节点数	迭代次数	总误差
2	10000	0.000949444	13	10000	0.001175002
3	10000	0.000055523	14	10000	0.000004807
4	10000	0.000030904	15	10000	0.000212522
5	8776	0.000105445	16	10000	0.000028057
6	10000	0.000012537	17	1473	0.000113128
7	733	0.000033940	18	10000	0.000222287
8	10000	0.000054020	19	10000	0.000020770
9	10000	0.000013394	20	1446	0.000000415
10	369	0.000000987	21	2784	0.000000999
11	10000	0.000018163	22	1585	0.000016172
12	4051	0.000022619	23	4565	0.000001000

由表2可知,隐含层节点数为20时的网络逼近精度最好,其次是节点数为10,再次是21、23。显然与本文提出的优化算法的仿真结果相吻合,这说明了本文的优化算法的应用效果是令人满意的,而且还具有试验次数少、精度高、简单、直观、有效、节省成本等优点。

结论 从黄金分割原理出发,提出的一种前向网络隐含层节点数的优化算法是切实可行的,不仅能搜索到理想的隐含层节点数,而且能起到节省成本、提高搜索效率等功效。

(上接第51页)

在的问题:失效节点个数增加将会导致坐标空间内碎片的产生,一些节点掌管了大量的“区”。为了处理这个问题,Can有一个节点重新定位的算法,用来合并那些可以合并的“区”。

4.3 结构化的文件查询算法优点

结构化的文件查询算法有很多优点:(1)效率高,目标的定位可以在确定的跳数内完成,节点间通信的平均延迟较小;(2)流量小,每次路由转发只选取一个节点,方向性强,不会像“Flooding”算法一样带来大量的网络流量;(3)扩展性好,节点路由表的大小和路由所需的跳数随着系统规模的扩大呈对数级增长,通信性能随着系统规模的扩大优美降级,具有良好的可扩展性;(4)可靠性好,节点之间不同路径数随系统规模的扩大呈对数级增长,如果路由路径中有节点失效,路由算法可以绕过失效节点,选择另一条路径,到达目的地,通信可靠性较高。

可以看出,结构化的P2P系统综合考虑了系统的可扩展性、通信的效率和路由可靠性,在三者之间作了有效的折衷,比前面介绍非结构化P2P系统更适合于构造大规模分布式存储系统。

总结与展望 上述一些算法是目前常见的P2P网络结构中所采用的文件查询算法。它们的优点和缺点反映出了设计者最初针对不同的网络结构特点所做的工作。根据现有文件查询算法的不足,我们提出了如下一些值得改进的方向。

非结构化的P2P:非结构P2P网络的特征就是存放数据的位置和网络的拓扑结构完全无关。我们在这种网络上查询文件时只能是随机地查找,如果不考虑收敛性问题,非结构的P2P网络对文件共享和开发应用来讲是一个较好的选择。

因此改善收敛性问题是非结构化P2P网络发展的关键和研究方向。

例如,采用“并行随机漫步”^[10]来替代“Flooding”洪泛算法,改善收敛性和文件检索的效率。

结构化的P2P:在结构化的P2P网络系统中,当我们为了访问数据而创建标识符时,主要存在如下两个问题:

1. 数据的局域性被破坏。由于文件是根据fileID来确定

算法中既充分考虑到能保证在隐节点数“频繁”出现的区间中搜索出理想的结果;又兼顾实际应用中高精度的需要,可将搜索区间做进一步拓展以求得逼近能力更强的隐含层节点数。

该优化算法简单、实用,具有很好的应用价值。

参考文献

- 1 Hornik K, Stinchcombe M, White H. Multilayer feedforward networks are universal approximators. *Neural Networks*, 1989, 2(5):359~366
- 2 Blum E K, Li L K. Approximation theory and feed forward networks. *Neural Networks*, 1991, 4(4):511~515
- 3 Huang S C, Huang Y F. Bounds on number of hidden neurons in multiplayer perceptrons. *IEEE Trans on Neural Networks*, 1991, 2(1):47~55
- 4 Sartori M A, Antsaklis P J. A simple method to drive bounds on the size and to train multiplayer neural networks. *IEEE Trans on Neural Networks*, 1991, 2(3): 467~471
- 5 Reed R. Pruning algorithm—a survey. *IEEE Trans on Neural Networks*, 1993, 4(5):740~747
- 6 靳蕃编著. 神经计算智能基础. 成都:西南交通大学出版社, 2000. 135~136
- 7 赵弘,周瑞祥,林廷圻. 基于Levenberg-Marquardt算法的神经网络监督控制. *西安交通大学学报*, 2002, 36(5):523~527

存放位置,这使得文件可能被保存在距离访问请求比较远的地方,数据的局域性(locality)受到一定的影响,这也就意味着本来在一个小范围内经常访问的数据,可能会被随机存放在外部空间,失去了预先定位、高效查询的机会。

2. 对应用层的信息不能合理利用。实际上,大量的网络应用会自然地分层描述文件数据,这说明了文件之间隐式存在着远近关系。但是结构化的P2P网络根据文件标识符来分配存储节点的算法,明显无法利用这种现有的关系来合理分配资源。

同时结构化P2P网络缺乏一定的灵活性(resilience),这是由于路由表被分散到了系统中的各个节点上,每个节点都要参与路由,当节点加入或者是离开网络时,有效地维护路由信息是比较困难的。

参考文献

- 1 The gnutella homepage: <http://gnutella.wego.com>
- 2 larke I, Sandberg O, Wiley B, Hong T. Freenet: A distributed anonymous information storage and retrieval system. In: Proc. of ICSI Workshop on Design Issues in Anonymity and Unobservability. June 2000
- 3 Stoica I, Morris R, Karger D, Kaashoek MF, Balakrishnan H. Chord: A scalable peer-to-peer lookup service for internet applications. In: Proc. of SIGCOMM, Aug. 2001
- 4 Rowstron A, Druschel P. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In: Proc. of the 18th IFIP/ACM Intl. Conf. on Distributed Systems Platforms, Nov. 2001
- 5 Ratnasamy S, Francis P, Handley M, Karp R. A scalable content-addressable network. In: Proc. of SIGCOMM 2001, Aug. 2001
- 6 Gummadi K, Gribble S, et al. The Impact of DHT Routing Geometry on Resilience and Proximity. In: Proc. SIGCOMM'03, Aug. 2003
- 7 Bosselaers A, et al. SHA: a design for parallel architectures. In: Proc. Eurocrypt'97, April 1997
- 8 Zhao B Y, Kubiatowicz J, Joseph AD. Tapestry: An infrastructure for fault-tolerant wide-area location and routing. Technical Report UCB/CSD-01-1141, 94720, April 2001
- 9 Saroiu S, Gummadi PK, Gribble SD. Exploring the design space of distributed peer-to-peer systems: Comparing the web, triad and chord/cfs. In: Proc. of the 1st International Workshop on Peer-to-Peer Systems (IPTPS'02), March 2002
- 10 Lv C, Cao P, ECohen, Li K, Shenker S. Search and replication in unstructured peer-to-peer networks. Preprint, Optional 2001