基于PNS-PGrid的P2P路由算法的设计与实现

宁多彪[1],陶中平[2],吕光宏[3]
(1. 成都东软信息学院,四川 成都 611844;
2. 成都理工大学,四川 成都 610063; 3. 四川大学,四川 成都 610065)

摘 要:目前绝大多数的P2P网络系统都是以覆盖网络方式构建的。在覆盖网络中相邻的节点在底层网络中可能并不相邻甚至相隔很远,这样导致覆盖网络中两个节点间会有很大的路由延迟。只有节点路由表项的内容正确地反映节点之间在底层网络中的拓扑关系,才能最终减少应用层的路由延迟,提高网络应用的性能。文中介绍了几种结构化P2P路由机制:Chord,CAN,Plaxton,Tapestry,Pastry和PGrid;以及几种非结构化P2P路由机制:Napster,BitTorrent,Gnutella和FreeNet。重点分析了PGrid路由算法。针对PGrid路由算法的路由表维护的盲目性和优化周期长等缺点,提出了一种新的基于邻近度选择技术的路由表维护算法PNS-PGrid(proximity neighbor selection PGrid)。PNS-PGrid是在节点转发一个查询请求后,触发路由表维护任务,并对本次转发使用的路由表项进行优化,且优化周期根据路由表项是否达到或接近最优值而进行调整。PNS-PGrid算法中还加入了对未报告的节点失效和异常退出的处理机制来对路由表进行维护。最后在开源软件PGrid中实现了PNS-PGrid算法。测试表明,PNS-PGrid算法在较少的开销下使路由表项能动态地有针对性地进行调整,并且快速地达到最优值,最终减少路由延迟,提高网络性能。

关键词:覆盖网络;PNS-PGRID;P2P;分布式哈希表
中图分类号:TP393 **文献标识码**:A **文章编号**:1673-629X(2009)12-0047-04

Design and Implementation of P2P Routing Algorithm Based on PNS-PGrid

NING Duo-biao[1], TAO Zhong-ping[2], LÜ Guang-hong[3]
(1. Chengdu Neusoft Institute of Information, Chengdu 611844, China;
2. Chengdu University of Technology, Chengdu 610063, China; 3. Sichuan University, Chengdu 610065, China)

Abstract: At present, most of the P2P network systems are constructed as overlay networks. Neighboring nodes in an overlay network are not necessarily close to each other or may even far away from each other in the underlying network, which may result in high routing delay between two nodes in the overlay network. Only if the entries in the routing table correctly represent the underlying network topology can the routing delay in the application level be reduced and the routing and network application performance be improved. Introduces several structured P2P routing mechanisms: Chord, CAN, Plaxton, Tapestry, Pastry and PGrid, and several unstructured P2P routing mechanisms: Napster, BitTorrent, Gnutella and FreeNet, and then focuses on the analysis of the PGrid routing algorithm. Aiming at the disadvantage of blind optimization, long optimization cycle and etc. of the maintenance method used in PGrid which selects routing table entries to optimize periodically and randomly, presents a new routing table maintenance algorithm called PNS-PGrid(proximity neighbor selection PGrid)). In PNS-PGrid, the routing table maintenance task is invoked after a node has forwarded a request and the entry used at this time is optimized. The optimization cycle is adjusted according to whether the routing table entry reaches or is close to the optimization value. PNS-PGrid also incorporates the mechanism to handle the unwarned node failure and exceptional departure to maintain routing table. Experiment shows that PNS-PGrid algorithm can dynamically and pertinently adjust the entries in a node's routing table under a comparatively low cost and can reach the optimization value faster, which finally reduces routing delay and improves network performance.

Key words: overlay network; PNS-PGRID; P2P; DHT

0 引 言

目前绝大多数的P2P网络系统都是以覆盖网络方式构建的。在覆盖网络中相邻的节点在底层网络中可能并不相邻甚至相隔很远,这样导致覆盖网络中两

收稿日期:2009-04-07;**修回日期**:2009-07-22
基金项目:国家自然科学基金(60802064)
作者简介:宁多彪(1969-),男,甘肃武威人,硕士,副教授,研究方向为分布式存储、网络性能分析。

个节点间会有很大的路由延迟[1]。只有节点路由表项的内容正确地反映节点之间在底层网络中的拓扑关系,才能最终减少应用层的路由延迟,提高网络应用的性能。高度动态的 P2P 网络有着复杂的拓扑结构,可以将 P2P 网络结构分为以下三类:非结构化的,结构化的,松散结构化的。而机构化和非结构化的根本区别在于每个节点所维护的邻居是否能够按照全局方式组织起来以利于快速查找[2]。

在非结构化数据(对象)的存放是与覆盖网络拓扑结构完全无关的。节点在存取数据的时候,不知道该数据的具体存放位置,只能通过一种"洪泛"(flooding)的机制发起随机的搜索请求,同时查询大量的节点,询问它们是否有符合查询条件的文件[3]。代表的系统主要有 Napster、Bittorrent、Gnutella。非结构化 P2P 系统因它们创建 Overlay 拓扑结构的方式和在节点间分发查询请求的方式的不同而不同。在该结构中节点的加入和退出是相对自由的,容易适应高度变化的节点群,缺点是想获得高的数据查获率,必须在系统中尽可能广地散播查询消息,因此非结构化 P2P 系统中的节点数量就要受到限制,扩展性较差[4]。

结构化网络主要是作为解决非结构化系统所面临的扩展性差这一问题而被提出的。在结构化网络中,对象 ID 和对象存放的节点间存在一个映射关系并以分布式路由表的形式提供,对象的存放位置可以被精确定位。因此查询请求可以被高效的路由到存放该数据的节点,避免了随机搜索的随意性。代表系统有 Tapestry、Chord、Pastry、CAN 等。结构化网络的缺点是不支持模糊对象匹配,从而使定位机制缺乏对结果的可选择性,且对于高度动态的 P2P 网络要维护其结构化拓扑需要一定的开销。而且在 Chord、CAN 中没有考虑覆盖网络结构和底层网络中实际距离的关联,导致了逻辑拓扑中的一跳可能对应与物理网络中的多跳的问题[5]。

1 PGrid 算法分析

分布式散列表 (Distributed Hash Table, DHT) 覆盖网络是一类新兴的 P2P 网络结构。在 DHT 系统中,对象和节点根据分布式散列算法(如 SHA-1),分别被赋予一个全局唯一的标识符 nodeId 和 ObjectId,对象依据自己的标识符,存放在相应的系统节点中[6]。

PGrid 是一种经典的基于 DHT 覆盖网络的分布式结构化 P2P 路由算法。对使用 PGrid 路由算法的 DHT overlay 系统来说,系统的局部性(locality)性质包括两方面:路由的局部性和副本定位的局部性。路由的局部性指一个请求能以最短的路径、最小的延迟从请求发起节点到达目的节点;副本定位的局部性指节点每次文件查询能找到距离自己最近的文件副本。而要实现路由的局部性,必须保持节点路由表的 proximity 性质[7]。经证明,若节点路由表中每个表项都是指向符合要求的在底层网络中距离自己最近的节点的话,可以使得路由延迟最小,且定位到最近的文件副本[8]。

PGrid 使用虚拟二叉树来构造 P2P overlay 网络,PGrid 中的节点位置由节点在虚拟二叉树中的路径(path)决定,如图 1 中节点 1 在 2 层的位置为 00,节点 2 的 2 层位置为 01,则节点 1 存放以 00 为前缀的对象索引,节点 2 存放以 01 为前缀的对象索引。每个节点含有一个路由表,其中每一路由项指向同层中一个与该节点在其某路径位上具有相反值的节点。

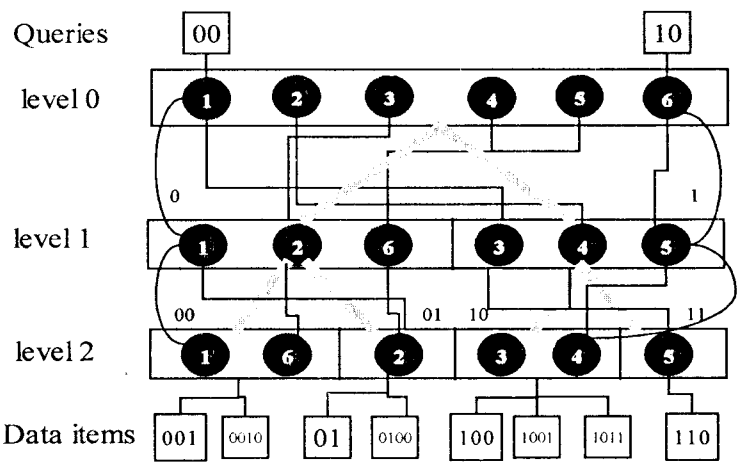

图 1 PGrid 虚拟二叉树拓扑结构

PGrid 的对象查询采用基于前缀的路由,例如当节点 1 收到一个对标识 00 的查询时,由于其本身负责以 00 为前缀的对象查询,故节点 1 完成标识 00 的查询,当节点 6 收到一个标识为 10 的查询时,由于其在 0 层对应的路由指针是节点 5,故将查询路由到节点 5,同理节点 5 将请求转给节点 4,直到查询成功或失败。PGrid 定位开销的时间复杂度为 $O(\log N)$。

PGrid 中原有的周期性随机选择路由表项进行维护的方法虽然也引入了邻近度邻居选择技术,但是这种的路由表维护算法存在以下几个缺点:

(1)盲目性:随机选取路由表中的路由表项进行优化的方式缺乏针对性,比较盲目。如果节点正好处于通向一个热门文件的路由路径上,这时路由表中的某项可能会被频繁地使用。而由于该方法随机选择要优化表项,可能使得该项不能得到及时的优化,将给路由性能带来负面的影响[9]。

(2)优化周期长:由于优化周期长度固定不变,路由表无法以最快的速度将自己调整到优化的状态。而在路由表经过一段较长时间的调整已经接近优化时,该方法仍然使用固定的周期实施优化操作,不断测量网络距离,这又将无谓地增加节点负载和网络开销。

(3)优化幅度较小:该方法从自己的路由表中随机的选取一项(设该项处在第 i 行),比较自己到该项路由表中第 i 行的各项的距离,如果距离更近,则用新的节点替换自己路由表中相应项的内容。这样每次的节点数都较少,优化幅度较小。

针对以上几个问题,提出了一种基于 PGrid 改进的 P2P 路由算法。

2　PNS－PGrid 算法的设计与实现

基于 PNS－PGrid(Proximity Neighbor Selection PGrid)算法的基本设计思想是节点在转发一个查询请求后触发路由表的维护任务,有针对性地对本次转发使用的路由表项进行优化。如果某节点正好处于通向一个热门文件的路由路径上,这时路由表中的某项可能会被频繁地使用,这种由转发驱动的路由表维护算法使该路由表项能够及时地优化,避免了随机选取路由表项进行优化的盲目性[10]。此外,一个路由表的某项经过数次优化之后将达到或者最大程度地接近最优值。这时再启动路由表的优化算法也难以进一步优化该项,反而会增加节点和网络的开销。为了避免这种情况,改进后的算法采取可变周期地运行路由表维护任务来不断地优化路由表项的邻近度(proximity)性质,这样也比定长周期的优化调整到优化状态的速度更快。同时为了判断节点是否活跃,并且减少失效节点在一定时间内被修复好后再次加入网络所带来的额外开销,该算法定期发送检测消息以检查节点是否活跃,并根据具体情况维护路由表,使查询消息能正确地被路由到目的地。

下面给出一个利用 PGrid 系统构建的通信实例。它对于验证工作的正确性以及进行性能分析具有重要作用。它主要完成的工作有网络节点的建立和加入,节点间的通讯,以及在此过程中路由表的维护。

```
public static void main(String args[]) {
// process the command arguments
Log.init(args);
doIinitstuff(args);
//create a DistHelloWorld object
DistHelloWorld driver = new DistHelloWorld();
// create first node
PGridNode pn = driver.makePGridNode(true);
// We wait till the first PGridNode on this host is ready so that the
rest of the nodes find a bootstrap node on the local host
synchronized (pn) {
while (! pn.isReady()) {
try {
pn.wait();
} catch (InterruptedException e) {
System.out.println(e);
}
}
// create other PGrid nodes
for (int i = 1; i < numnodes; i++) {
pn = driver.makePGridNode(false);
synchronized (pn) {
while (! pn.isReady()) {
try {
pn.wait();
} catch (InterruptedException e) {
System.out.println(e);
}
}
}
if (Log.ifp(5)) {
System.out.println(numnodes + "nodes constructed");
}
// send message to the random selected node
Random rng = new Random();
for (int i = 0; i<nummsgs; i++) {
for (int client = 0; client < driver.helloClients.size(); client +
+) {
HelloWorldApp app = (HelloWorldApp) driver.helloClients.get
(client);
app.sendRndMsg(rng);
}
}
}
}
```

3　性能分析

3.1　路由性能测试

为了测试算法改进后路由表维护算法对较大规模系统环境下路由性能的改变,通过编写仿真程序,模拟了一个有 1000 个节点的广域网环境中的 PNS－PGrid 测试系统。模拟系统中的 $b=4$,$|L|=16$。

在 DHT 覆盖网络中,文件查询消息是通过在覆盖网络的节点间转发实现文件定位,这使得消息沿 overlay 路由路径走过的距离要比源节点和目的节点在底层网络中的直接距离要长,用“latency stretch”表

示这两个距离之间的比值,它的值永远大于等于 1。

latency stretch 是反映 DHT 路由表 proximity 性质的重要一个指标。从前面的分析可以知道,在 DHT 覆盖网络中,如果节点路由表能正确反映节点之间在底层物理网络中的距离关系,具有很好的 proximity 性质时,latency stretch 值将较小;反之,latency stretch 值将变大,路由延迟增加。

对实验系统从初始状态开始,分别使用周期性随机选择路由表项优化的方法和 PNS－PGrid 算法进行节点路由表的优化操作。在系统运行过程中,节点随机地发起文件查询请求,记录系统的平均 latency stretch 值随系统运行时间的变化趋势。

图 2 是系统的 latency stretch 值随运行时间的变化趋势。从图中可以知道,在系统初始阶段,由于各节点刚建立起自己的路由信息,此时路由表还不能很好地反映节点在底层网络的距离情况,proximity 性质不好,所以 latency stretch 值较大,路由延迟很高。随着系统的运行,节点路由表不断得到优化,系统的 latency stretch 值逐渐下降。最后经过一段时间的优化,节点路由表被优化到接近最优状态,latency stretch 趋于稳定。

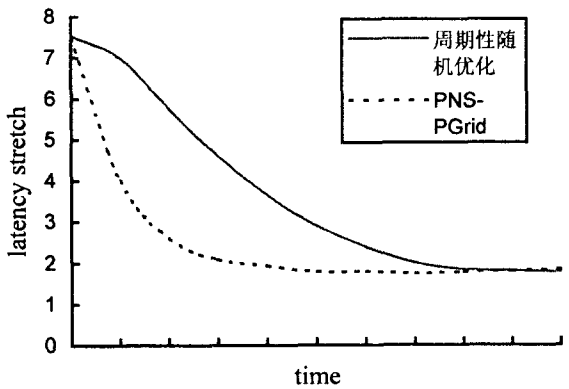

图 2 latency stretch 随时间变化图

从图中可以发现两种算法的优化性能的差异。周期性随机优化算法由于采用周期性随机选择路由表项进行优化,优化效果逐步累积, latency stretch 变化曲线下降速率相对稳定,路由表达到最优状态的速度较慢。而对 PNS－PGRID 算法来说,在系统运行初期,节点每次转发都将触发对路由表的优化操作,针对性强,且每次都测量较多节点,优化幅度较大。节点路由表以较快的速度被调整到最优值,表现在 latency stretch 曲线上呈前陡后缓的变化趋势,比使用周期性随机调整的算法提前达到稳定状态。

3.2 优化开销测试

为了不断地优化路由表的 proximity 性质,节点需要不断地测量自己到某些节点的距离,以发现更接近自己的节点,替换路由表中相应项。将节点发起的对其他系统节点的距离测量操作次数作为衡量路由表优化算法的系统开销的指标。通过记录一个新加入系统的节点距离测量操作次数随系统运行时间的变化情况,对周期性优化算法和 PNS－PGrid 两种优化算法的系统开销进行了比较,见图 3。

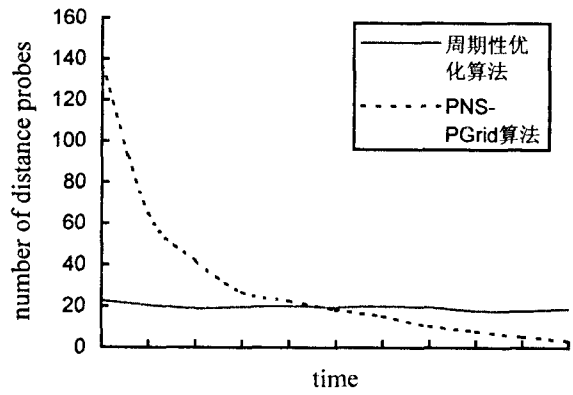

图 3 优化开销随时间变化图

从图中可以知道,周期性维护算法的优化开销在文件的整个生命周期里保持相对稳定,且维持在较低的水平。而 PNS－PGrid 算法在节点加入系统的初期为了达到快速优化节点路由表的目的,每一次路由转发都会触发优化算法,且每次需要测量的节点数都比较多,所以优化开销较大。但是,初期优化开销增加带来的好处是优化效果明显,节点路由表迅速被调整到最优状态。根据 PNS－PGrid 算法的设计,节点每次测量的相邻节点数目将随着优化次数增加而减少,故优化开销曲线下降迅速,很快就降到与周期性优化算法同样的水平。在路由表达到最优状态后,PNS－PGrid 优化操作启动的频度将逐渐降低,优化开销曲线继续下降,低于周期性优化算法。到后期,PNS－PGrid 只需要极少的开销即可保持节点路由表最优。

4 结束语

对 P2P 的相关知识进行了简要介绍,然后介绍了 P2P 网络结构以及几种结构化和非结构化的 P2P 路由机制,接下来,对 PGrid 路由算法进行了深入的分析和研究,提出了一种新的基于邻近度的路由表维护算法“PNS－PGrid”。该算法通过动态有针对性地调整节点路由表项的内容,使之获得更好的 proximity 性质,使 P2P 网络的逻辑拓扑结构和物理拓扑结构更加匹配,减小路由延迟,获得更好的网络性能。最后在开源软件 PGrid 中实现了 PNS－PGrid 算法。

(下转第 54 页)

```
e.ord=curstep; e.seat.i=i; e.seat.j=j; e.di=di;
push(s,e);
.....
pop(s,e);
back(kmq,e.seat.i,e.seat.j,e.di);//撤棋
curi=e.seat.i;curj=e.seat.j;curdi=e.di+1;
```

在两个算法中,进栈部分的代码是相同的,栈 s 以清晰的 e(ord,seat,id)格式存储着移棋步骤。而出栈的代码则不同,由于栈 s 在递归算法中存储的移棋步骤信息正好也是回溯点所需保存的信息,所以本算法在非递归转化过程中,并不需要再增加新的栈,但此时栈 s 已经具有两个作用:一是提供撤棋函数 back( )的参数以及恢复现场变量 curi、curj、curdi 的值,实现棋局回溯;二是保存移棋步骤序列。通过这一比较,就很好理解"递归算法非递归化一般都是借助栈来实现"这一设计方法了。

非递归算法不使用系统栈,利用递归与循环算法的内在规律,借助程序栈,将程序向循环转化,算法执行过程不依赖于函数或过程的重复调用,大大降低了时间复杂度。本算法所需辅助空间只是程序栈 s 的存储空间,空间复杂度降低了。

非递归算法的缺点是程序复杂且难以分析和理解。因此在求解实际问题时可以采用递归思想来分析,然后用非递归来实现算法[8]。

5 结束语

文中针对孔明棋求解方式的特点,以栈为数据结构,利用回溯方法实现了递归和非递归算法,实验结果验证了方法的有效性,对同类问题的算法设计具有一定指导和参考意义。本算法还可在多方面进行研究和探讨,如求解孔明棋的所有解、探索路径的优化等。

参考文献:

[1] Ford W,Topp W. Data Structures with C++[M].[s.l.]: Prentice Hall,1996.

[2] 严蔚敏,吴伟民. 数据结构(C 语言版)[M]. 北京:清华大学出版社,2006.

[3] 辛 玲,王相海. 国际象棋中马的周游路线问题的递归算法[J]. 计算机工程与设计,2006,27(1):47-48.

[4] 李立中,林顺喜. 孔明棋的电脑解法研究[EB/OL]. 2002-08-23. http://alg.ice.ntnu.edu.tw/icewise/docs/Kon-Ming.doc.

[5] Kruse R,Tondo C L,Leung B. Data Structures & Program Design in C[M].[s.l.]:Prentice-Hall,1997.

[6] Sahni S. Data Structures Algorithms and Applications in C++[M]. Columbus:Mc Graw-hill,1999.

[7] 杨庆红,罗 坚. 递归问题的非递归实现方法研究与应用[J]. 计算机时代,2005(8):44-45.

[8] 马军红. 递归与非递归算法比较及效率分析[J]. 科技信息,2008(31):554-556.

(上接第 50 页)

测试表明,PNS-PGrid 算法在系统运行初期,节点每次转发都将触发对路由表的优化操作,针对性强,且每次都测量较多节点,优化幅度较大,而后节点路由表能以较快的速度被调整到最优值,并且达到优化开销较小的状态。

参考文献:

[1] 杨文俊. P2P 网络系统中节点自组织管理机制[J]. 计算机技术与发展,2006,16(7):57-59.

[2] Stoica I, Morris R, Karger D, et al. Chord: A scalable peer-to-peer lookup service for Internet applications[R].[s.l.]:MIT,2001.

[3] Bolosky W, Douceur J, Ely D, et al. Feasibility of a serverless distributed file system deployed on an existing set of desktop pcs[C]// Proceedings of SIGMETRICS 2000. Santa Clara,CA:[s.n.], 2000.

[4] Waldman M, Rubin A D, Cranor L F. Publius: A robust, tamper-evident, censorship-resistant, web publishing system[C]// Proceedings of the 9th USENIX Security Symposium (August 2000). Berkely: IEEE Press,2000:59-72.

[5] Aberer K. P-Grid: A self-organizing access structure for P2P information systems[C]// Proc of IFLP International Confrerence on Network and Parallel Computing Workshop. Los Alamitos: IEEE Computer Society Press, 2007:437-441.

[6] Sandberg R. The Sun Network Filesystem: Design, Implementation and Experience[C]// Proceedings of the 1987 Summer Usenix Conference. New York: ACM Press,1987:300-314.

[7] Oceanstore project home page[EB/OL]. 2006. http://oceanstore.cs.berkeley.edu. BLACK M.

[8] Druschel P, Rowstron A. PAST: A large-scale, persistent peer-to-peer storage utility[M]. HotOS VIII, Schoss Elmau, Germany:[s.n.], 2001:75-80.

[9] 张庆丰,李东琦,唐慧佳. 基于 P2P 分布式文件传输系统的研究[J]. 微计算机信息,2007,23(3):92-94.

[10] 赵慧娟,王汝传. 基于遗传算法的 P=2P 资源发现算法[J]. 南京邮电大学学报:自然科学版,2007,27(4):85-89.