

基于 Freepastry P2P 算法与实现

A new P2P algorithm base on freepastry

邬家炜 彭少斌 李锋 (华南师范大学计算机学院 广州 510631)

摘要: P2P 为防止网络失败,必须及时有效定位在线资源,发现技术成为 P2P 关键技术。本文基于 freepastry 路由算法实现思想和原理,并在此基础上改进其中不足,最后通过实验比较验证,改进后新算法确能明显提高 P2P 对资源发现、定位和查找效率。

关键词: P2P 发现技术 DHT 网络 聚集度 短链

对等网络(P2P)打破传统 C/S 模式,每结点在网络中地位都是对等的,既充当服务器,同时也享用服务。它为个人用户提供便利与自由的同时,也试图整合网络资源。然而,拓扑结构繁而复杂,难以兼容,加之节点频繁加入和离开,如何高效定位有效资源成为近几年研究热点。

1 Pastry 和 Freepastry 简介

Pastry 是微软提出的可扩展布式对象定位和路由协议^[1]。发现算法能有效检索查找资源,并能保证搜索跳数在 $O(\log N)$ 范围之内。但它对节点拓扑结构和存储位置限制过多,不符 P2P 发展趋势。Freepastry 是基于 Pastry 发布的软件包,采用 PNS-CG (Constrained gossiping Proximity neighbor selection) 启发策略定位物理邻居节点,降低 pastry 因采用 PNS 启发策略导致的复杂性,拥有更好的查找效率和适应性。

2 Pastry、Freepastry 和新算法区别

Pastry 主要实现 DHT 功能,而 Freepastry 实现基于下层路由的 DHT。对比两套路由算法,其本质区别在于下一跳路由选择方式: Pastry 直接使用标准算法^[2]选择下一跳,而 freepastry 在标准算法基础上选择最小 proximity metrics 节点作为下一跳。然而,在实际网络拓扑中,既不存在非结构化系统所认识的完全随机图,也不是 DHT 发现算法采用的确定性拓扑结构,而是一种具有高聚集度和短链特性^[1]的网络

拓扑。我们知道,聚集度较低的结点与其它结点联系较少,通过该结点查找效率偏低;反之,聚集度越高查找效率亦随之提高。当网络中存在大量高连通结点时,因节点聚集度差异导致节点间存在短链现象。因此,产生和寻找短链对下一跳路由选择方式举足轻重。经此分析,笔者提出一种基于 nodeId 优化的 DHT 发现算法,与 Freepastry 不同之处在于增加一个最亲近 ID 号选择函数,使其优先于 proximity metric 路由选择函数进行。在路由信息选择下一跳时,新增函数将待查信息和路由表项众节点比较,优先选择与待查信息最为接近的节点作为下一跳,从而找出短链,实现对当前层路由 nodeId 路由优化,有效缩短路由跳数。流程图如图 1。

3 设计与测试

新算法采用 JAVA 语言编写,源程序选用 Freepastry 1.4.2 版本,开发平台是 JDK1.4.2。通过对标准路由器源码分析,其路由选择用 receiveRouteMessage(RouteMessage) 函数实现。因此修改相应的 RouteSet,增加 mostSimilar 函数,用来从 RouteSet 对象中获取 ID 最接近节点。

一方面为方便试验,减小网络规模, nodeId 取 32bit, $b=2$, $L=2 * 2b=8$ 。另一方面为便于与 freepastry 算法比较,节点间不通过网络而采用直接连接,使用模拟器模拟多节点网络拓扑结构。网络模拟器使用欧几里德空间模拟网络,网络节点间距离满足三角不等式。

实验建立 3000 个节点,从中任意选择节点发送 30000 条随机信息。图 2 给出单节点在传递信息过程

proximity metrics 最近的作为下一跳。无论表项有多大,所选下一跳一般不会改变,因此路由路径没有太大改变,节点路由信息数也无明显变化。而新算法随 routesets 增大明显降低路由信息数,这是由于新算法选择与待查信息最为接近的 nodeid 作为下一跳,利用短链有效缩短路径,因此路由信息明显减少,而且随着 routesets 的增加,路由信息数降低速度放慢。

图 3 给出一条信息平均被路由的跳数,横坐标同前,纵坐标表示路由跳数。此图与上图相似,二者相互关联,亦可相互导出。引起原由是 freepastry 采用 proximity metrics 优化下一层路由,和 routesets 大小无关,routesets 的扩大并不能带来明显好处。新算法采用 nodeid 优化当层路由,扩大 routesets 会增加找到短链机会,从而缩短路径。但随着 routesets 上升,跳数减少渐缓,预计在 routesets 某点路由跳数减小将微乎其微,这也表明仅靠扩大 routesets 路由表项缩短路径是有限的。新算法本质上是采用一种空间换取时间的方法,通过路由表项中保存的局部节点信息减少路由时间,但同时,不断扩大路由表项会带来更多 nodeid 查找的复杂度。减少和增加相互抵消,二者逐渐达到平衡,在平衡点上路由效率达到最高。

图 4 给出发送单条信息到目标节点所经历的 proximity metrics 值。从图中可以发现两点:第一,路由表项 routesets 对信息 proximity metrics 几乎没有什么影响,仅是路由表从单表项到多表项变化剧

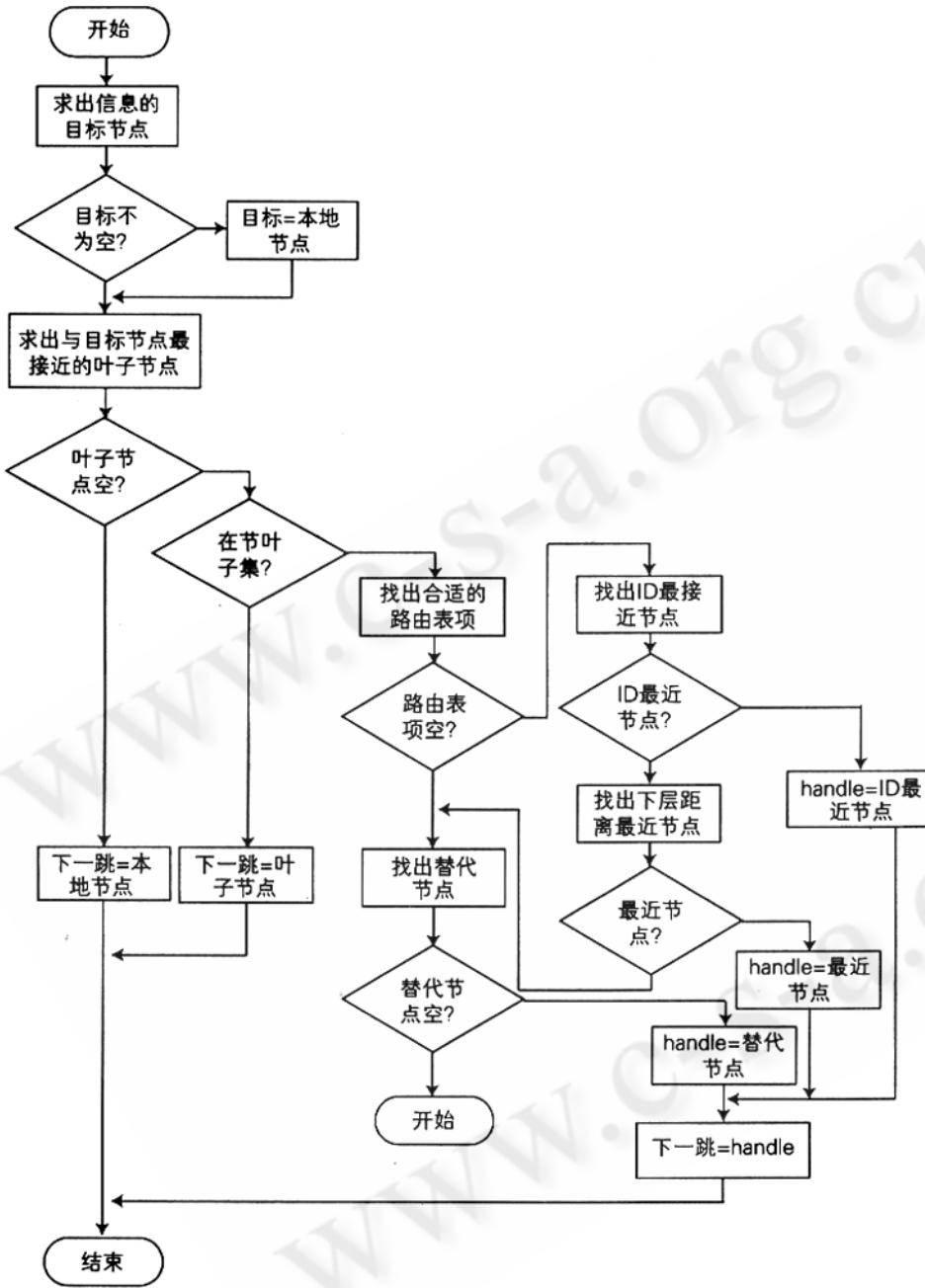


图 1 新算法流程图

中所使用的路由表路由平均信息数。横坐标 (routesets) 表示路由表中单个表项能够包含节点的句柄上限,纵坐标 (numbers) 表示路由信息数。从图中可以看到,随着 routesets 从 1 逐步增加至 8, freepastry 单节点路由信息数几乎没有任何变化。这是由于 freepastry 仅是从相应路由表项中选取与当前节点

由效率达到最高。

图 4 给出发送单条信息到目标节点所经历的 proximity metrics 值。从图中可以发现两点:第一,路由表项 routesets 对信息 proximity metrics 几乎没有什么影响,仅是路由表从单表项到多表项变化剧

烈,往后渐变平稳。单表项到多表项跃变可能是由节点状态维护引起的。第二,新算法无论是完整路径上还是在单跳过程中,所经历的 proximity metrics 值都比相应 freepastry 要大,这与各自下一跳选择策略有关。图五描述单跳 proximity metrics 值,同样 proximity metrics 对 routesets 几乎是常量,而新算法单跳则明显高于 freepastry。两图可以推知,虽然新算法信息所经 proximity metrics 值普遍高于 freepastry,但相差不大,而且在第 8 点还低于 freepastry。究竟是何种原因使新算法既可维持较少当前层路由跳数,又兼顾到下一层 proximity metrics 呢? 这问题需从两方面解答。

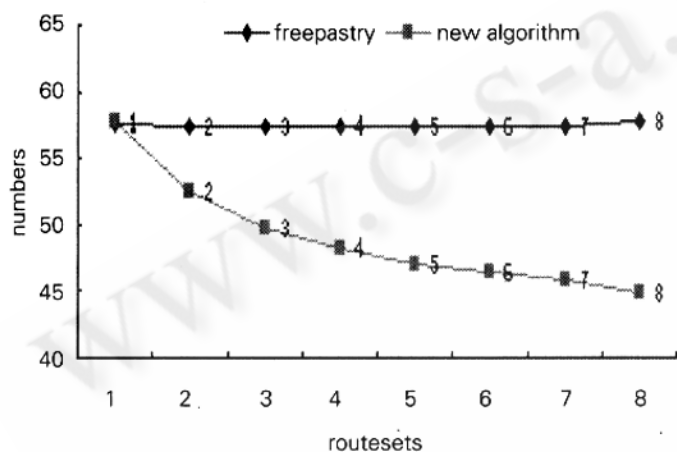


图 2 单节点路由信息数

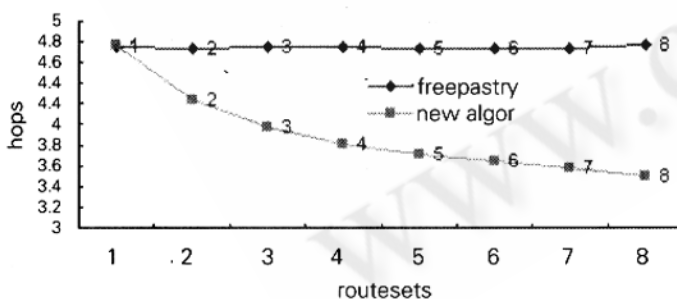


图 3 单条信息路由跳数

(1) 图六给出路由算法在选择下一跳时,选择最近 nodeld 和最近 proximity metrics 之间的比例。可以清晰看到,超过 98% 的下一跳节点选择最近 nodeld,这也证明新算法确实选择到网络中的短链。而在这 98% 选择中,最近 proximity metrics 又占据一定比例(一般在 30% 以上),因此在新算法路由中,下

一跳最近的 proximity metrics 值约为 $98\% \times 30\% \approx 30\%$,如图七所示。正是因为拥有这 30% 比例使 proximity metrics 值保持较低。

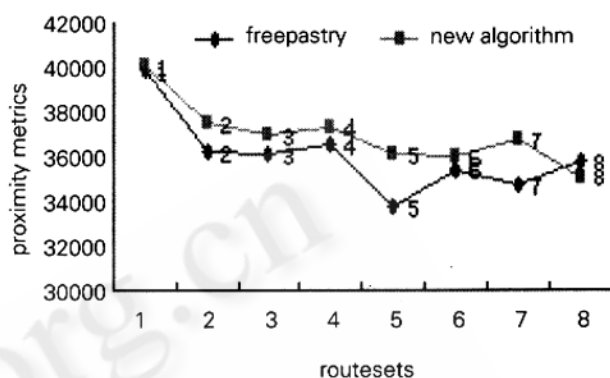


图 4 单条信息经历的 proximity metrics 值

(2) 从前文对当前层路由优化讨论可知,虽然新算法选择下一跳节点的平均 proximity 值都明显大于 freepastry,但由于当前层路由跳数减少,因此单条信息 proximity metrics 也会相应的减少。

综合上述两方面原因,新算法可以在优化当前层路由的同时保持较小 proximity metrics 值,而且当掌握足够局部信息之后能找到比 freepastry 更小的 proximity metrics 路由。

4 总结与展望

从试验结果可以得出,基于 nodeld 优化的新发现算法能利用短链现象,有效缩短路由跳数。由于它优先考虑搜索层,在网络层路由效率可能不及 freepastry,但随着搜索层路由跳数的缩短,发现效率明显提高。

P2P 技术发展到今天,新应用软件和模式层出不穷,并倍受业界关注。从集中目录式的 Napster,到完全分布式 Gnutella,以及动态集群式 BitTorrent, P2P 应用呈现更多的分布化,网络组织管理呈现更好的适应性和动态性,资源定位、查找和共享也更高效充分。本文提出的发现算法,既顺应 P2P 发展趋势,又提高了全分布式结构化拓扑网络资源的定位、查找效率,具有实际应用价值。不足之处是,新算法存在短链利用效率不高的问题,但无疑其思路是一个很好的开端。进一步工作将着眼于更高效的利用短链特性,以期达到更高发现效率。同时也启发我们更精确地了解

Internet 网络拓扑结构, 毕竟发现算法效率的提高很大程度上依赖于网络拓扑。

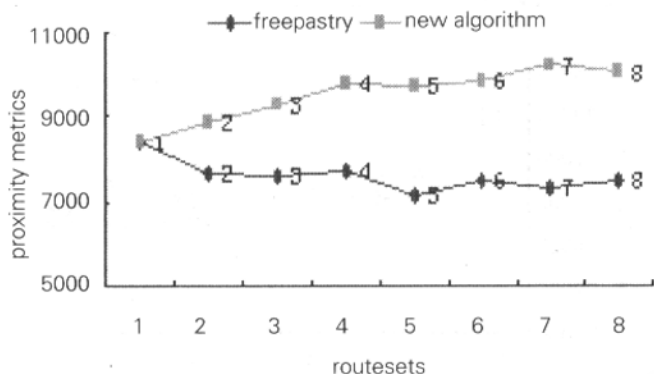


图 5 单跳的 proximity metrics 值

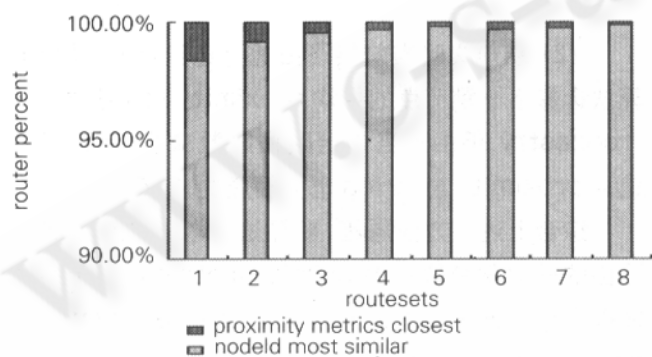


图 6 Most Similar 与 closest 的比例

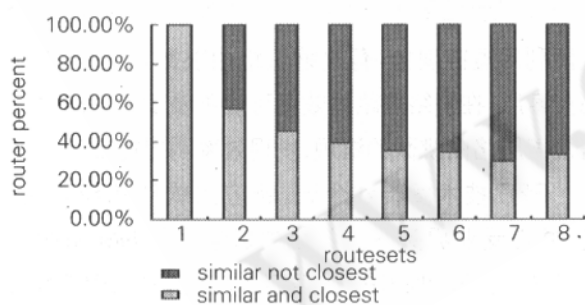


图 7 Most similar 中 closest 的比例

参考文献

- 1 吕向辰, P2P 技术与应用计算机世界, 2002 12 (9): 13-14.
- 2 Munindar P Singh. Peering at Peer-to-Peer Computing [J]. IEEE Internet Computing, 2003-01 /02.

- 3 Duncan J. Watts and Steven H. Strogatz. Collective dynamics of small-world networks. Nature. 1998. 393-440-442.
- 4 Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the internet topology. In SIGCOMM, 1999 pages 251-262.
- 5 M. Castro, P. Druschel, Y C. Hu, and A. Rowstron. Topology-aware routing in structured peer-to-peer overlay networks. 2002. Submitted for publication.
- 6 L. Gong. Peer-to-peer networks in action. IEEE Internet Computing, vol. 5, no. 1, 2002, pp. 37-39.
- 7 Steven D. Gribble, Eric A. Brewer. Scalable, Distributed Data Structures for Internet Service Construction.
- 8 J. Kubiawicz, D. Bindel, YChen, and B. Zhao. Oceanstore: An architecture for global-scale persistent store. In Proc. ASPLOS 2000, Cambridge, MA, November 2000.
- 9 S. Rathasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A scalable content-addressable network. In Proc. ACM SIGCOMM01, San Diego, CA, Aug. 2001.
- 10 P. Druschel and A. Rowstron. PAST: A large-scale, persistent peer-to-peer storage utility. In Proc. HotOS VIII, Schloss Elmau, Germany, May 2001.
- 11 I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for Internet applications. In Proc. ACM SIGCOMM01, San Diego, CA, Aug. 2001.
- 12 C. G. Plaxton, R. Rajaraman, and A. W. Richa. Accessing nearby copies of replicated objects in a distributed environment. Theory of Computing Systems, 32:241-280, 1999.