An Improved Routing Algorithm Based on Kademlia

LIANG Guang-min
The School of Electronics & Information Engineering，Shenzhen Polytechnic, Shenzhen 518055
gmliang@oa.szpt.net

Abstract: Because of the shortages of Kademlia in searching efficiency and the strategy of cache, this paper mainly studied on an improved Kademlia routing algorithm. By using fast table look-up and weights setting method, the improved Kademlia routing algorithm can accelerate the hotspots inquiry speed and congregates hotspot resources for high query hit rate in P2P network. Simulation tests on the PlaneSim platform show that the improved Kademlia routing algorithm has lower average route step and inquiry time delay than the original one.

Key words: P2P; Routing Algorithm; Kademlia

一种优化的 Kademlia 路由算法

梁广民
深圳职业技术学院电子与信息工程学院，深圳 518055
gmliang@oa.szpt.net

摘　要：由于 Kademlia 算法在查询效率和高速缓存策略方面存在不足，所以本文对该算法进行了改进。通过使用快表技术和加权设置策略，优化的 Kademlia 路由算法可以加快在 P2P 网络中热点查询的速度，同时为更高的查询命中率提供了更集中的热点资源。在 PlaneSim 平台上模拟实验证明了这种优化的路由算法实验查询延时更短、平均路由条数更少

关键词：P2P；路由算法；Kademlia

1 引言

随着网络的发展，尤其是大规模多媒体的开发和应用，在传统的 C/S、B/S 的结构中带宽和服务器的处理能力成为了瓶颈。相比之下，P2P 网络具有较高的自主性、可伸缩性、可靠性和对称性。尤其是可伸缩性在 P2P 网络中变成了研究热点。怎样高速和有效的查询资源节点是一个非常重要的问题。P2P 路由算法根据网络的组织结构、数据的分布和路由、定位方式主要分为两类：结构化的 P2P 和非结构化的 P2P。

结构化的哈希表（DHT）主要被用在结构化的 P2P 网络中。它是由在广域空间内分部的纵多分散的节点组成的一个大的哈希表。这个哈希表被分成众多不连续的部分。每个节点通过加密的哈希函数和统一的域名空间管理各自的区域。同时结构化的哈希表（DHT）还具有可伸缩性、可靠性、鲁棒性和自组织的特点。同时每个节点可以动态的加入和退出网络。目前，典型的结构化的 P2P 路由算法包括 CAN，Chord，Pastry 等等。

Kademlia 是一种基于 DHT 的路由算法。和前边提到的其它的路由算法相比，Kademlia 是一种新型的 DHT 覆盖拓扑结构，并且在查询速度上有明显的几个优点。它的特别之处在于利用 XOR metric 测量两个节点之间的距离，然而在查询效率和缓存策略上存在着不足。针对这种情况，本文引用了一种新的策略，该策略采用了快表技术和加权设置策略。这样就避免了在高的刷新频率下缓存查询效率低的问题，为更高的查询命中率提供了更集中的热点资源。

文章的组织如下：第二部分介绍了原始的 Kademlia 算法和它的数据结构；第三部分是改进的 Kademlia 路由算法；第四部分通过实验来证明该算法的有效性；最后一部分为结论。

2 Kademlia 算法描述

项目基金：深圳市科技计划项目，项目编号：QK 200608
Fund: Shenzhen Municipal Science and Technology projects (QK 200608)

978-1-935068-31-0

Kademlia 协议最早出现在 2002，美国的 PetarP. Maymounkov and David Mazieres 发表了“Kademlia: 基于 XOR metric 的点对点的信息系统”的文章引起了人们的重视。随后 2005 年，基于 DHT 的 Kademlia 路由算法首次运用在 BiTtorrent 中。之后，该算法在很多 P2P 的网络中得到运用，如 BitComet，BitSpirit 和 eMule，它们所采用的获取值、节点和关键字的算法不同。

在 Kademlia 算法中，每个节点都有一个长度为 160 位的字符串作为唯一标识，这 160 个字符是通过随机函数选取自由组合而成的。同时，每个节点都有一个<关键字，值>的一个值对（但并不是必须的），值存储在最近的节点上。为了使这些值处于活跃状态，需要周期性的发布。

Kademlia 哈希表的组织形式为一个二叉树，叶子为 Kademlia 节点。根据节点 ID 的最短的唯一前缀给每个节点赋值，每个节点根据其 ID 插入树中的某一位置。从树根开始，每一位代表一个二叉树的分支。每个节点都拥有整个二叉树的一部分空间，除了自身之外，其余部分又被划分成一系列连续的子树。最高子树是除了根节点之外的整个二叉树的一半，下一个子树是剩下部分的一半，等等，以此类推。节点 0010 如图 1 所示。

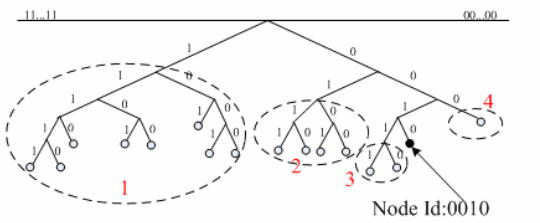

Figure 1. Node subtree

图 1. 节点子树

根据子树的结构，XOR 度量被用于计算两个节点之间的距离。举例，节点 x 和 y 之间的距离用表达式 $d(x, y) = x \oplus y$ 表示，从该表达式中我们可以看出，高的数字比低的数字有更多的影响。正如我们所知道的：XOR 度量是单向的，每个 x 都有唯一的 y 和唯一的 d(x，y)和其相对应，搜索相同关键字会沿着同样的路径进行，该路径独立于起点。如果我们在该路径上只存储<关键字，值>对的话，这种方法能减轻热点区的压力，提高查询效率。

在 Kademlia 中，任何节点和它周围的节点都保持 logN（N 为节点数，logN 默认值为 160）个连接。此外，每个节点都有一个特殊的路由表，被称为 k-bucket 表，该表和其它节点的三元组（IP 地址，UDP 端口，节点 ID）至多包含 K 个入口。同时，在覆盖的节点中，为了使路由更加牢靠，通过跨越几个不相连的路径，参数 K 是一个冗余因子。例如 k-bucket[i]，节点之间的信息被存储，节点之间的距离在 2i 和 2i+1 之间。图 2 显示了 K=20 时，k-bucket 表的情况。

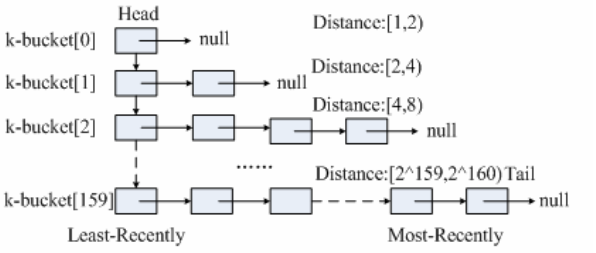

Figure 2. k-bucket table

图 2. k-bucket 表

总之，在 Kademlia 算法优点如下：节点 ID 之间的关系，在 k-bucket 中节点和节点位置之间的距离都被定义。

然而，高速缓存策略有一些缺点。首先，要成功查询<关键字，值>时，没有必要去存储这个值。第二，产生和搜索资源降低了效率，尤其是热点节点。如果资源是无效了，关于它的信息在每一个相关的节点上将会被清除，因此，热点节点的资源要在缓冲中重新建立。

3 改进的 Kademlia 路由算法

为了改进以上存在的问题，本文提出了一种改进的 Kademlia 算法，该算法主要通过使用快表和加权设置来提高热点区查询的效率。

3.1 高速缓存策略

由于最初的 Kademlia 算法不能区分热点节点和普通节点，新的缓存策略提出，在每个节点中除了现存的 k-bucket 外，在空间允许的基础上，要增加一个新表用于存储存在高访问率的节点信息。在本文中，该表被称为快查询表或快表，快表的每个记录包括资源和供应者即<关键字，值>对。每个节点的查询时间也被记录在表中，因此热区的存储能力和缓冲能力都被提高了。

热区由 counter 的值来定义，例如，我们定义 counter>3 的资源为热区，因此，检查每个节点的 counter，如果大于 3，则将该节点存储到邻居节点或者有最靠近 key 值的节点。注意，热区对于邻居节点

来说意味着要把热区资源存储在表中。拥有最靠近的 key 的节点的存储操作的含义是发表该 key 的信息。

3.2 更新快表

更新快表就犹如在 Kedemlia 路由模型中更新 k-bucket，当一个新的记录被加入到快表中，步骤如下:

（1）如果该资源已经在快表中存在，counter+ = 1，然后把指针移向尾部

（2）如果资源不存在，并且记录的数目少于 k，把资源添加到快表的尾部

（3）如果表是空表，则删除表顶部的记录，然后在表的尾部增加一个新记录。

4 仿真实验

本文中，运用 PlanetSim 来运行和评估 Kademlia 路由算法。PlanetSim 是一个网络模拟器，用于大规模 overlay 服务的实验框架。它提供了一个统一的方法来模拟，并且可以清晰的区分算法的设计以及基于之上的应用和服务。因为节点是随意的插入 PlanetSim 平台上，算法的有效性可以保证。PlanetSim 提供了一系列的评估指标，如路由步长、查询时间延迟、查询成功次数和失败次数等等。

在本文中，该模拟实验主要通过评估路由步长、查询时间延迟等指标来比较新旧算法关于热区查询的问题。查询数增加，参数不变。

首先考虑平均路由步长，单位是一个跳变。图 3 显示了平均路由步长和查询数量的关系。“Orgn Kad”表示原始的 Kademlia 协议，而“New Kad”表示改进的算法。结果证明，改进的算法没有显著增加系统的负担，但可以通过增加一个合适快表的方法来改善热区搜索。

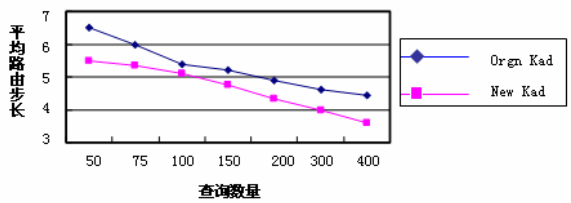

Figure 3. Average routing steps

图 3. 平均路由步长

同样的环境下，查询时间延时也被检测。查询时间延时被定义为平均时间延时，单位是毫秒。图 4 表明，y 值的变化依赖于 x 轴的值，通过 y 轴可以看出，随着查询数目的增加，查询时间延时会减少。统计学数据证明，改进的 Kademlia 算法在减少查询时间延时方面具有显著性。

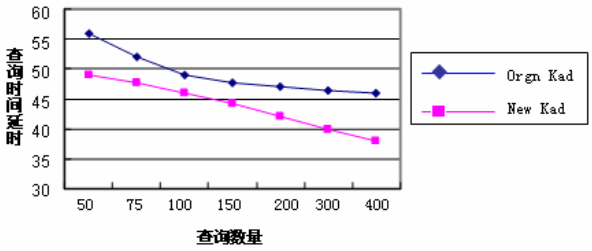

Figure 4. Query time delay

图 4. 查询时间延时

5 结束语

本文重点关注的问题是 P2P 网络中的如何高效的发现节点和检索资源的问题。重点分析了 Kademlia 算法和改进的 Kademlia 路由算法。在真正的 P2P 网络中，利用快表查询和加权设置方法可以加快热区的查询速度。实验结果证明，改进的算法具有较低的路由步长和查询时间延时，同时可靠性也得到了提高。

References (参考文献)

[1] Yang B, Garcia-Molina H,Improving search in peer-to-peer networks. In: Sivilotti PAG, ed. Proc. of the Int' l Conf. on Distributed Computing Systems[C]. IEEE Computer Society, 2002.

[2] Kong Yu, Zhang Sheng, Liu Huaping, the Kademlia DHT-based Routing Algorithm [J], West China University of Technology (Natural Science), 2009 (6).
孔玉,张升,刘华萍, 基于 DHT 的 Kademlia 路由算法改进[J], 西华大学学报（自然科学版）,2009(6).

[3] Liang Guangmin, An Improved Kademlia Routing Algorithm for P2P Network[C],2009 International Conference on New Trends in Information and Service Science,2009.

[4] Ding Yanyan, WAN Zhen-kai, based on the Kademlia search strategies such as network research [J], instrumentation users, 2007 (6).
丁艳艳,万振凯,基于对等网络的 Kademlia 搜索策略的研究[J], 仪器仪表用户,2007(6).

[5] tong-Qing Qiu, Gui-Hai Chen, a P2P overlay network topology so that a common approach [J], Journal of Software, 2007,18 (2)
邱彤庆,陈贵海,一种令 P2P 覆盖网络拓扑相关的通用方法[J], 软件学报,2007,18(2).

978-1-935068-31-0 © 2010 SciRes.