

基于 DHT 的 Chord 路由算法改进

宗平¹, 徐 鸽²

(1. 南京邮电大学 海外教育学院, 江苏 南京 210046;

2. 南京邮电大学 计算机学院, 江苏 南京 210046)

摘要: Chord 路由查找算法是基于 DHT 散列表的, 在可扩展性、稳定性及负载均衡方面有一定优势。鉴于网络中有些节点配置较低, 导致 Chord 性能降低。文中为了改善通过低性能节点的概率, 针对 Chord 本身的单路查找过程, 采用起始点的并发式查找定位, 通过多路径来定位资源节点, 以此达到避免通过性能差的节点的可能, 改进性能, 减少查询时延。通过实验进行 Chord 查询时延和查询跳数的性能分析, 仿真结果表明, 在保持 Chord 原有健壮性的基础上, 改进算法能够有效提高查找效率和降低查找时延。

关键词: Chord; 分布式散列表; 路由查找

中图分类号: TP301.6

文献标识码: A

文章编号: 1673-629X(2012)09-0139-04

Improvement of Chord Routing Algorithm Based on DHT

ZONG Ping¹, XU Ge²

(1. College of Overseas Education, Nanjing University of Posts & Telecommunications, Nanjing 210046, China;

2. College of Computer, Nanjing University of Posts & Telecommunications, Nanjing 210046, China)

Abstract: Chord used DHT as a search strategy base. It has the advantages in load balancing, reliability, scalability and others. However, when there are some low efficiency nodes in the network, Chord will be affected with the low efficiency and cause high latency terribly. So it proposes a parallel routing finding model beginning with the start node against Chord's single routing mechanism. This improved algorithm decreases the probability of routing with bad nodes and then increases the finding efficiency. Analyse the performance of finding latency and hops by experiments. And maintaining the robustness of the original Chord, the simulation experiments show that the improved algorithm can enhance the efficiency and reduce the time delay of the routing finding.

Key words: Chord; DHT; routing finding

0 引言

当前, P2P 网络技术早已运用于文件共享、即时通信、流媒体、共享存储和对等计算及协同处理等技术领域, 对于 P2P 网络技术方面的相关研究也已是国内外学术界的一个重点。相比于传统的 C/S 服务模式, P2P 网络的一个显著特点就是节点间都是对等关系。对等网络理论研究必须解决的核心问题无疑是如何高效准确地实现网络资源的查找。所以在对 P2P 网络中资源查找的研究就成为如何进一步更好地提高 P2P 网络服务质量的关键。目前, 该领域的资源查找算法基本上有四种: 以 Gnutella^[1] 为代表的非结构化分布式查找算法, 以 Napster^[2] 为代表的集中式查找算法,

以 Chord^[3] 为代表的结构化分布式查找算法以及以 Kazaa 为代表的混合式查找算法。结构式分布网络中由于并不存在服务器这一角色, 因而基于此的查找算法更加符合 P2P 网络的要求。相比于非结构化的分布式查找算法, 结构化分布式查找算法^[4] 由于没有采用泛洪算法, 从而拥有较强的系统可扩展性, 研究价值也更高。

文中基于对传统的 Chord 算法进行技术分析, 以此来引入一种改进的新型 Chord 算法。仿真实验结果表明, 改进算法能够有效地提高查找资源的效率和降低查找的时延。

1 Chord

1.1 关键字查找

Chord 协议是一种由 MIT 提出的 P2P 网络路由协议。对于处于 Chord 中关键字和节点, 分配一个大小为 m 比特的相应的标识符。通过一致哈希函数^[5] 来得出 Chord 中关键字的标识符 Key, 并表示为 m 位标识值,

收稿日期: 2012-01-04; 修回日期: 2012-04-10

基金项目: 江苏省科技支撑项目 (BE2009157)

作者简介: 宗平 (1956-), 男, 博士, 教授, 研究方向为计算机网络、物联网技术等; 徐鸽 (1985-), 男, 硕士研究生, 研究方向为计算机在通信中的应用。

并以哈希节点的 IP 地址为基础来计算出节点标识符 id。该算法中采用 SHA-1 作为散列函数。对网络中的节点的标识符排序,然后从小到大以顺时针方向围成一个标识环,该环即为 Chord 环。Chord 环上存储着节点标识各不相同的关键字,关键字以 K(Key, Value)表示。其中后继节点表示为 successor(K),也就是在 Key 的顺时针方向上与其距离最近的节点。

在 Chord 环中的节点都存在一张最多包含 m 个表项的路由表需要更新维护,即指针表^[6](finger table)。对于节点 n 的查找表来说,其中的第 i 个表项包括 $s = \text{successor}(n + 2^{i-1})$, $1 \leq i \leq m$ 而且对其中每个都要计算 $\text{mod}2^m$,用 s 来表示节点 n 的第 i 个指针,即 $n.\text{finger}[i]$ 。

- Chord 查询过程如下:
- (1) 当节点接收到查询请求后,先计算关键字 K 是否在对应的节点标识及其后继节点区间内;如果在区间内,则查找成功;否则转(2);
 - (2) 节点继续在 Finger 表中搜索,并将表项中小于 K 的最大节点标识符中的最后一个节点定位,向该节点转发此查询请求;
 - (3) 反复上述过程,最终能够锁定到 K 的后继节点,即目标 (K, V) 对的存储节点。

图 1 给出了 N8 的路由表及查找 K53 的过程。

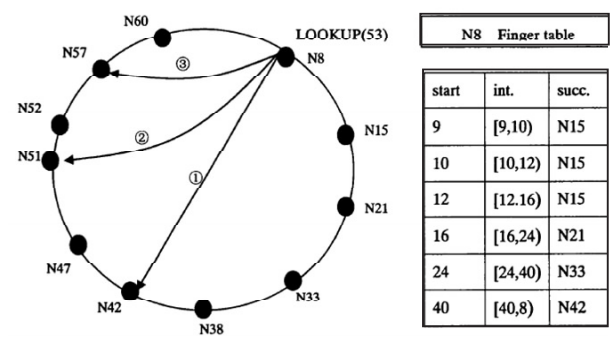


图 1 Chord 的节点路由表及查找

1.2 节点的加入与退出

当新节点 n 加入时,Chord 必须完成以下三个步骤:

- (1) 定义新节点的前驱及路由表项,并通过某已知节点查找新节点 n 指针表中各表项;
- (2) 对已存在节点的前驱及路由表进行更新,通过调用相关函数,完成指针表内容的更新;
- (3) 告诉其后继节点将应该由 n 负责的资源关键字索引传递给 n 。

- 节点的离开处理:
- (1) 某节点请求退出时,必须将所有存在该节点的路由表更新,以其后继节点来代替;
 - (2) 系统中的节点都将拥有一张由 r 个最近后继

节点组成的列表,来保证在节点离开后系统的查询服务仍然得以进行。并将以列表中的第一个节点来代替离开节点。

1.3 Chord 算法的不足

- (1) 低性能节点造成的系统瓶颈。
节点性能差异的问题将随着网络中节点数量的扩张而变得突出,此时低性能节点必将阻碍 Chord 查找进度,从而引起系统瓶颈^[7]。因为在 Chord 协议系统中,不管各节点之间性能差距多大,其承担的责任都是一致的,如查询、存储资源等。低性能节点的出现将导致系统性能低下、请求响应时延增大等。
- (2) 节点加入或离开问题。

由于 P2P 网络是实时变化的,每时每刻都有发生着节点加入或离开,而在节点加入或离开后系统都需要更新原节点所属的路由信息表,以此来保证表中信息的准确性。比如说以一个 N 个节点的 Chord 系统为例,某一节点的加入或离开,要通过 $O(\log_2 N)$ 次数据交换来更新 Chord 路由信息以及分配相关的文件。当系统中的节点加入或离开过于频繁,导致整个网络的资源将消耗在更新节点路由信息及文件重建上,从而浪费大量的系统带宽,降低系统的查询效率。

2 Chord 算法的改进

基于 Chord 环的简单构造和以分布式散列算法的特点,可以将环中的对象空间对应到相应的虚拟空间,通过对虚拟空间中对象的均衡分布,以此来实现对象到节点的映射。Chord 系统的结构化,相比其他非结构化系统,该系统在空间与时间复杂度及路由唯一性都有显著优点。

2.1 优化查找策略

- Chord 资源定位^[8]步骤可以概括为:
- (1) 计算查询关键字 key 的 hash,SHA-1 (key);
 - (2) 将得到的 hash 进行比较,搜索本机中是否存在 key 的 hash,若存在,则查询,否则转(3);
 - (3) 对本地 finger 表查找最接近又小于关键字的 hash 节点;
 - (4) 然后转发查找请求给(3)中得到的节点,重复(2)至(4)步,直到查询终止。

对以上步骤分析可以得出,对于在请求点自身计算机中完成的步骤(1)至(3),其时间复杂度是一定值,所以可以知道查询产生的时延是由第(4)步的转发查询请求造成的。因为对于 Chord 环中的节点映射,未能对节点自身性能、响应效率及节点通信情况给予充分的考虑,当 P2P 网络中的节点大量增加后,必将存在一些性能比较底下的节点,所以当转发的请求信息经过这些节点时,会导致查询产生较大的时延^[9]。

因此,改进的关键点之一就是对于当前的查找策略进行优化,来避免请求查询转发消息通过低性能节点,以避免由于低性能节点而导致的查找时延^[10]。

2.2 并发式 Chord

基于对 Chord 的分析,能够得出以下结论:Chord 查找是当一个请求节点发送一个查询请求后,等到该请求被回复后才继续发送下一跳查询请求,如此反复直到成功找到目标节点为止的单线的迭代式查询^[11]方法。文中提出一种通过请求节点同时发送多条请求查询消息来对目标节点进行定位的机制,即并发式 Chord 路由机制,在该机制中,从相关节点中选取返回响应请求最快的作为下一跳节点,能够避免因通过低性能节点而导致的系统查询性能低下的问题。

文中给出的优化方法并不会对传统 Chord 的节点路由由表映射结构进行改变^[12],只是在查询过程中进行系统优化。不失一般性,文中针对并行发送 3 条请求(3S-Chord)为例,说明改进的查询步骤:

- (1) 计算出查询关键字 key 的 hash,即 SHA-1(key);
- (2) 对得到的 hash 进行比较,来判断该 hash 是否在本机,若在则直接查找,否则转(3);
- (3) 从本机的 finger 表中找出 3 个不大于该 hash 的又最接近于它的节点;
- (4) 将查询请求转发给步骤(3)中得到的那 3 个节点,反复进行(2)至(4)步,直至查找成功。

采取并发式 Chord 机制优化后的每一步查询过程中,通过性能低下的节点概率相比原系统降低了 2/3。

图 2 以 N8 查找 K53 为例说明了 3S-Chord 的查询过程。

- 步骤一,N8 对当前的 finger 表进行搜索,在当前表项中并没有找到 K53,则跳到步骤二;
- 步骤二,采用对当前目标节点与当前 finger 表的对比,从而得到 3 个最接近的节点 N42、N33、N21;
- 步骤三,同时发送 3 条请求查询 lookup(53) 消息到 3 个最邻近节点。

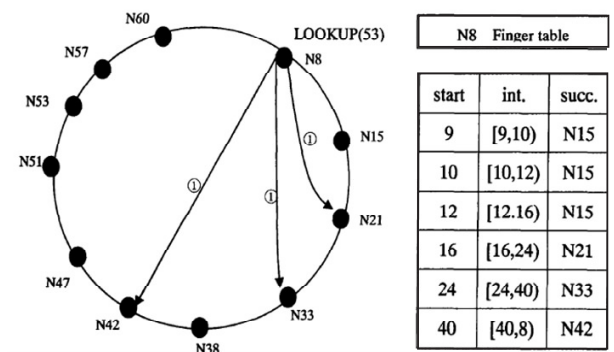


图 2 3S-Chord 资源查找

下一跳节点来自于返回响应速度最快的节点,反

复循环,以成功查找到节点 K53 为止。

假设 N33 为网络中存在的低性能节点,那么在查找的过程中系统将选取其他两个节点来完成节点定位,以此实现本系统的设计目的,即低时延、高效率。

从上述对 3S-Chord 的分析与研究,能够增加节点数量,直至 n 条并发,即 NS-Chord。从理论上讲,网络中通过低性能节点的可能性与请求数成反比,即当请求并发数为 n 时,概率即可降为原系统的 $1/n$,然而在实际环境中,当网络中节点数量过多时会造成网络负载压力增大。所以,对 NS-Chord 中 n 的选值应该要根据不同网络而决定。

3 仿真及分析

文中采用 P2PSim 软件在 Linux 操作系统下进行仿真。为了能够让仿真环境更加接近真实性,在此将节点数选取为从 100 ~ 1000 的区间。在此条件下,将优化后的协议与原 Chord 协议进行性能比较,比较参数是系统的平均查询跳数及平均查询时延。

3.1 3S-Chord 性能分析

在平均查询时延的分析过程中,分别对于 3S-Chord 和原 Chord 进行 10 组实验,并进行对照分析。图 3 中为 3S-Chord 和 Chord 的查询时延对比,以实验节点数作为 X 轴,平均查询时延(ms)作为 Y 轴。从图中可以发现,随着节点数的增加,3S-Chord 的平均查询时延明显优于原 Chord 的,而且随节点增加的变化规律仍符合 $O(\log_2 N)$ 复杂度。

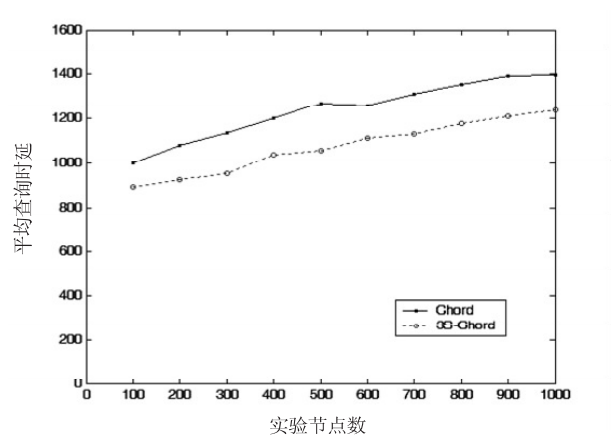


图 3 3S-Chord 和 Chord 的查询时延对比

图 4 显示的是平均查询跳数的比较情况。从图中可以看出改进后的算法查询跳数相对于原 Chord 有所降低。由于在 3 路并发查询过程中,系统接收到返回响应消息最快的节点后将同时终结其他尚未返回响应的查询消息,所以改进的算法并不会使系统的平均查询跳数增大。

3.2 NS-Chord 性能分析

在此设定网络节点数为 500,并修改并发查询路

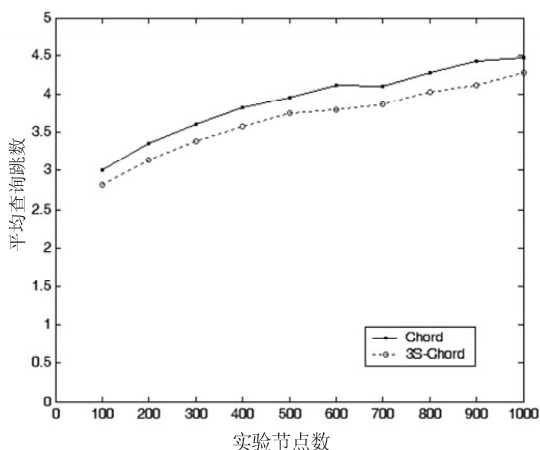


图 4 3S-Chord 和 Chord 的查询跳数对比数,来分析并发路数对于协议性能的影响。进行 10 次仿真,得到如下分析结果。从图 5 中可以发现,当并发数不大于 5 时,查询时延有比较明显的下降,但当并发数进一步增加时,查询时延的下降变得相对缓慢。说明当并发路数较多时,发送多路查询的过程及返回响应消息必将造成一定的时延,当并发路数过大时,网络复杂度的增大也将影响查询效率。

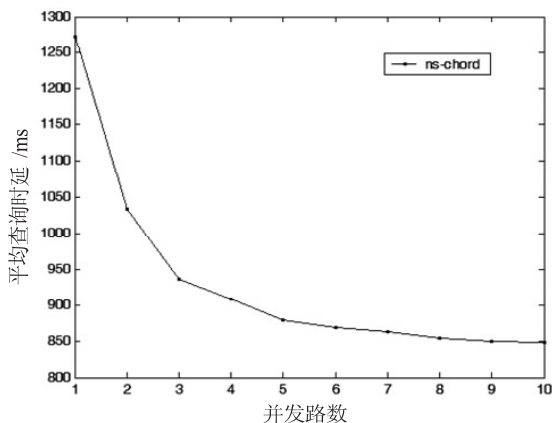


图 5 节点数为 500 时平均查询时延随并发路数的变化

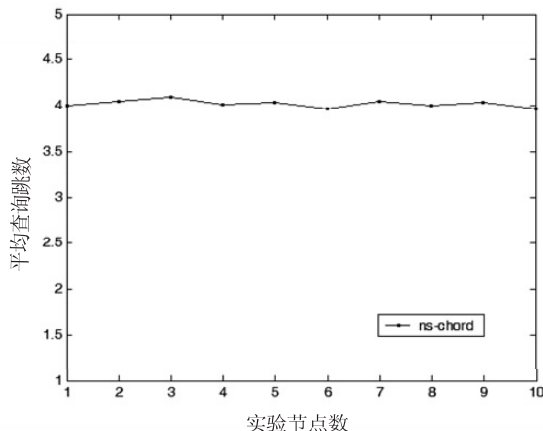


图 6 节点数为 500 时平均查询跳数随并发路数的变化

图 6 表示了平均查询跳数随着并发路数的变化情况。从图中可以看出,改进算法的平均查询跳数并不以当前并发路数的增加而改变明显,其稳定性能较高。因此改进后的算法保证了原协议查询质量。

4 结束语

文中在原 Chord 协议基础上给出了一种改进的并发 Chord 算法,有效地降低了查询时延和提高了查询性能。仿真结果证实,以多路并发查询为基础的并发 Chord 很大程度上避免了查询过程中通过低性能节点的可能性,在降低查询时延的同时,也改善了系统的查找效率。系统的改进并未造成查询跳数的过度增长,保证了原 Chord 的查询质量。

参考文献:

- [1] Ripeanu M. Peer-to-peer architecture case study: gnutella network [C]//First International Conference on P2P Computing. [s. l.]: [s. n.], 2001: 99-100.
- [2] Napster-file sharing system [EB/OL]. 2002-11. <http://napster.com/>.
- [3] Stoica I, Morris R, Karger D, et al. Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications [C]//Proceedings of ACM SIGCOMM 2001. New York, USA: ACM Press, 2001: 149-160.
- [4] 孙名松,周三山,刘杰. 分布式后缀树在 P2P 搜索中的应用 [J]. 计算机技术与发展, 2011, 21(3): 30-35.
- [5] 王继敏,宋玉蓉,蒋国平. 基于消息网络的 Hash 函数构造 [J]. 计算机技术与发展, 2011, 21(9): 24-27.
- [6] Spulber D F. Reputation in Auctions: Theory and Evidence from eBay [J]. Journal of Economics and Management Strategy, 2006, 15(2): 353-369.
- [7] 徐玉,程春玲,周芸. 基于树环 Chord 的大规模覆盖网的拓扑结构 [J]. 计算机技术与发展, 2011, 21(8): 97-100.
- [8] 万川龙,桑军,向宏,等. 基于 P2P 网络的视频点播系统设计 [J]. 计算机技术与发展, 2009, 19(6): 17-21.
- [9] Statistics of One-way Internet Packet Delays [EB/OL]. 1995. <http://www.ietf.org/proceedings/02mar/slides/ippm-4.pdf>.
- [10] Zhang Hui, Goel A, Govindan R. Improving lookup latency in distributed hash table systems using random sampling [J]. IEEE/ACM Transactions on Networking, 2005, 13(5): 1121-1134.
- [11] 张维凤,张代远. P2P 网络中基于文件路由模型搜索方法的改进 [J]. 计算机技术与发展, 2006, 16(12): 111-113.
- [12] 宁多彪,陶中平,吕光宏. 基于 PNS-PGrid 的 P2P 路由算法的设计与实现 [J]. 计算机技术与发展, 2009, 19(12): 47-51.

基于 DHT 的 Chord 路由算法改进

作者: [宗平, 徐鸽](#)
作者单位: [宗平\(南京邮电大学 海外教育学院, 江苏 南京 210046\), 徐鸽\(南京邮电大学 计算机学院, 江苏 南京 210046\)](#)
刊名: [计算机技术与发展](#)
英文刊名: [Computer Technology and Development](#)
年, 卷(期): 2012(9)

本文链接: http://d.g.wanfangdata.com.cn/Periodical_wjtz201209037.aspx