

文章编号:1001-9081(2006)03-0531-03

基于局部性的分布式哈希表资源定位技术

王芳¹,高玲¹,郑明春^{1,2}

(1. 山东师范大学 信息管理学院, 山东 济南 250014;

2. 中国科学技术大学 计算机科学技术系, 安徽 合肥 230026)

(sdsdwangf@126.com)

摘要:提出了一种基于局部性的分布式哈希表资源定位技术,将非结构化对等网络引入 Chord 中,充分利用节点在物理网络上的邻近性和节点间兴趣的局部性。模拟测试表明,该技术在路径长度和访问延迟方面的性能优于原 Chord。

关键词:资源定位;分布式哈希表;Chord;兴趣局部性

中图分类号: TP393.07 **文献标识码:** A

DHT resource location technique using locality

WANG Fang¹, GAO Ling-li¹, ZHENG Ming-chun^{1,2}

(1. College of Information Management, Shandong Normal University, Jinan Shandong 250014, China;

2. Department of Computer Science and Technology, University of Science and Technology of China, Hefei Anhui 230026, China)

Abstract: A core problem in large-scale Internet peer-to-peer application systems is the efficient location of the node that stores desired resource. A DHT resource location method using interest-based locality was presented. This method introduced unstructured P2P to Chord, making the best use of the proximity of peers in underlying physical network and the locality of the interest among peers. Simulation tests show that this method is superior to original Chord at path length and access latency.

Key words: resource location; DHT(Distributed Hash Table); Chord; interest-based locality

在传统的客户机/服务器网络中,以一些大的网站为中心,不同地域的客户端通过连接服务器进行信息的浏览和下载。随着这种网络的扩大,服务器已难以满足日益增加的客户端的需求,一旦服务器不堪重负而崩溃,整个网络也会随之崩溃。

对等网络(P2P)是没有中央控制和分级机构的分布式网络系统,由若干互联协作的节点构成,每个节点都同时是服务器与客户机,在网络中地位对等。这种网络结构可以充分利用网络上的资源,解决传统 C/S 网络中的服务器瓶颈问题。由于对等网络中的资源分布在各个节点上,且每个节点都可以随时进入或离开对等网络,因此如何高效地定位到所需资源是 P2P 系统面临的一个挑战。

1 问题的提出

目前 P2P 系统的资源定位技术基本上可以分为三类。第一类基于中央服务器(或服务器群),如 Napster^[1] 系统,中央服务器维护系统中所有共享资源的位置索引,由中央索引服务器提供查询和定位。该技术的优点是高效、易于实现,但存在单一故障点、可扩展性差等问题。另外两种技术将共享资源的索引分散在系统中的节点上,根据资源定位机制是否依赖特定的系统拓扑结构,分为非结构化和结构化定位技术。非结构化定位技术,如 Gnutella^[2],以洪泛方式将定位请求发送给自己的邻居节点,直到满足查询或超时,其优点是没有单一故障点,但存在带宽消耗大、可扩展性差等问题。结构化定位技术基于分布式哈希表(Distributed Hash Table, DHT),网络中的每个节点和资源经过 hash 后都有唯一的标识符 Id,节

点根据自己的 Id 负责一部分标识符空间,资源根据自己的 Id 映射到相应的节点;节点维护一个路由表,定位资源时通过路由表进行选择转发,在一定的跳数内定位到资源。近年来,各国研究人员在支持大规模对等网络的分布式查找服务方面进行了大量的工作,提出了各种结构化分布式查找系统,如 Chord^[3]。这类技术解决了非结构化定位机制的可扩展性问题,但由于没有考虑节点物理位置上的关系,只是优化节点间在逻辑网络上的跳数,使查找跳数较大,访问延迟较长^[4]。

研究表明^[5,6],当前的 P2P 系统都具有地域上的邻近性和兴趣上的局部性。对于节点 A,如果它在地理位置上靠近节点 B 且以后可能会为 B 提供服务,那么 A 和 B 存在地域上的邻近性,即地域上的局部性。A 的邻居节点上如果有节点 A 请求的资源,与距离 A 更远的节点相比,请求更容易到达节点 A。如果节点 C 在过去一段时间内响应了节点 A 的请求,节点 C 可能以后还会满足 A 的其他请求,那么 A 和 C 具有相似的兴趣,即兴趣上的局部性。由于 P2P 系统中提供相同服务的各点是平等的,节点可能会忽视一些与其“邻近”的、即通信效率较高的点,而与那些“遥远”的、相互之间通信效率不高的点进行数据交换。当前的资源定位机制由于没有全面利用 P2P 系统中内在的两种局部性,无法同时实现搜索高效和可扩展性。

本文面向 Internet 规模的文件共享、协同计算等 P2P 应用,对 Chord 进行改进,提出一种基于 Chord 的查找服务系统,在结构化 P2P 的基础上建立一个基于兴趣的非结构化覆盖网络,充分利用底层物理网络节点的邻近特性和节点间兴趣的局部特性。模拟测试表明系统在访问延迟和路径长度方

收稿日期:2005-09-15 修订日期:2005-12-11 基金项目:国家自然科学基金资助项目(60273041)

作者简介:王芳(1975-),女,山东枣庄人,硕士研究生,主要研究方向:对等网络;高玲(1980-),女,山东潍坊人,硕士研究生,主要研究方向:异质网络的性能评价;郑明春(1964-),女,山东济南人,教授,主要研究方向:对等网络、无线网络。

面的性能优于原 Chord, 同时具有 Chord 的固有优点, 如负载均衡、可扩展性等。

2 基于兴趣的资源定位

2.1 Chord 中的资源定位

Chord 使用相容哈希为系统中的每个节点和关键字产生唯一的标识符 Id, 如 IP 地址为 120.10.11.1 的节点经过哈希之后得到的标识符为 51, 而关键字“Letitbe”的标识符为 54。关键字保存在它的后继节点 (successor) 中, 后继节点是标识符大于或等于关键字 k 的第一个节点, 记为 $\text{successor}(k)$ 。如果标识符采用 m 位二进制数表示, 并且将从 0 到 $2^m - 1$ 的数排列成一个圆圈, 那么 $\text{successor}(k)$ 就是从 k 开始顺时针方向距离最近的节点, 如图 1 中关键字“Letitbe”存放在 $\text{successor}(54)$, 即节点 N56。如果 m 是关键字和节点标识符的位数 (采用二进制表示), 那么每个节点只需维护一张最多由 m 个表项组成的路由表, 称之为指针表。节点 n 的指针表中第 i 个表项包括的是 $s = \text{successor}(n + 2^{i-1})$, 这里 $1 \leq i \leq m$, 并且所有的计算都要 $\text{mod } 2^m$, s 称为节点 n 的第 i 个指针, 用 $n.\text{finger}[i].\text{node}$ 表示, 指针表中的其他项的含义如表 1 所示。图 1 中节点 N8 的指针表内容如表 2 左栏。进行资源定位时, 节点根据请求的关键字 Id 和指针表, 将查询请求转发到节点所知距离请求关键字最近的节点上, 实现高效查找。

表 1 指针表的结构

符号	定义
$\text{Finger}[k].\text{start}$	$(n + 2^{k-1}) \bmod 2^m, 1 \leq k \leq m$
.interval	$[\text{finger}[k].\text{start}, \text{finger}[k+1].\text{start}]$
.node	第一个大于等于 $n.\text{finger}[k].\text{start}$ 的节点
.successor	标识符环中的下一个节点; $\text{finger}[1].\text{node}$
.predecessor	标识符环中的前一个节点

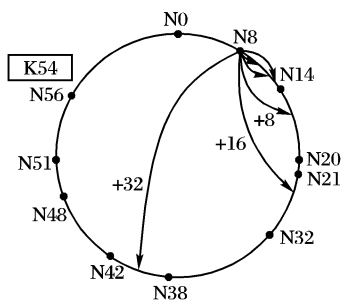


图 1 Chord 示意图

表 2 节点 8 的指针表

start	interval	node	start	interval	node
9	[9, 10)	14	9	[9, 10)	14
10	[10, 12)	14	10	[10, 12)	14
12	[12, 16)	14	12	[12, 16)	14
16	[16, 24)	20	16	[16, 24)	20
24	[24, 40)	32	24	[24, 40)	38
40	[40, 8)	42	40	[40, 8)	51
successor	14		successor	14	
predecessor	0		predecessor	0	

2.2 基于兴趣的资源定位算法

为了利用地域上的局部性, 反映节点之间在物理网络上邻近特性, 采用邻近邻居选取 (Proximity Neighbor Selection, PNS) 的方法改进 Chord 的指针表: 节点 n 的第 i 个指针不再

是第一个大于等于 $n + 2^{i-1}$ 的节点, 而是从标识符介于 $n + 2^{i-1}$ 和 $n + 2^i$ 的节点中选择前 x 个节点 (可能之间的节点数少于 x); 然后根据物理网络上的邻近原则, 从这些节点中选择距离 n 最近的节点作为节点 n 的第 i 个指针。例如图 1 中, 节点 N32 和 N38 同处于节点 N8 的指针表第 5 个指针的标识符跨度, 如果节点 N38 比节点 N32 距离节点 N8 更近, 那么使用 PNS 方法改进后, 节点 N8 的第 5 个指针即为节点 N38, 而不是 N32。若对节点 N51 和 N42 进行比较后有同样的结果, 那么节点 N8 改进后的指针表如表 2 右栏。

本文提出的资源定位技术基于兴趣的局部性^[7], 即从与自己具有共同兴趣的节点处更有可能找到所需资源。我们称具有共同兴趣的节点为朋友节点, 节点 n 的所有朋友节点构成了节点 n 的兴趣社区。为了构建兴趣社区, 节点首先要确定自己的朋友节点。判断节点 X 和节点 Y 是否是朋友的一种方法是基于历史数据, 即在过去一段时间内节点 X 是否能够满足节点 Y 发出的请求。本文使用另一种方法, 基于当前数据判断两个节点是否是朋友, 即通过两个节点的当前共享内容判断节点间兴趣的相似程度。

节点的兴趣社区可以看作是一张图, 顶点是节点, 当两个节点间有共同兴趣即共享了相同的文档时, 在图中相应的两个顶点间建立一条弧。每条弧可以根据不同的测度设置权重, 本文使用两个节点共享相同文件的个数作为弧的权重。创建社区的节点称种子节点, 第一步, 种子节点找到网络中与其共享了相同文档的节点。具体做法是: 种子节点随机地选取本地存储文档的子集, 向网络中的一些节点发送请求, 查看这些节点是否存储了这些文档; 请求发送的对象从节点所知道的加入了 P2P 系统的节点集合中选取; 与种子节点至少有一个相同文档的节点被加入到图中, 这些节点的深度为 1, 即它们与种子节点直接相连。第二步, 对于由第一步加入到图中的每个节点 p 重复第一步, 新的节点也被加入到图中, 这些节点的深度为 2。在节点 p 和新加入的节点 q 之间建立弧, 其权为两个节点间相同文档的个数。第三步, 在图中加入一个收点, 并与所有节点连接, 通过求解种子节点和收点之间的最大流问题确定种子节点的朋友节点, 并建立了种子节点的兴趣社区。所有节点的兴趣社区构成了一个非结构化的对等网络。图 2 是节点 N8 和节点 N42 的兴趣社区, 虚线表示深度为 2 的朋友节点。

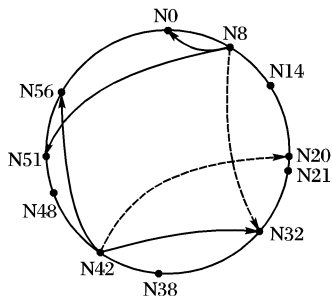


图 2 节点 8 和节点 42 的朋友节点示意图

算法描述如下:

- 1) 种子节点开始创建兴趣社区, 种子节点的深度为 0。
- 2) 对于深度 $i = \{0, 1\}$ 的每一个节点 p : 从 p 的本地文档中随机选取 k 个文档, 这些文档的关键字构成集合 r ; 从 p 所知道的节点中随机选取 n 个节点, 向这些节点请求 r 。

对于至少满足一个文档的每个节点 q : 将 q 加入到图中, 其深度为 $i + 1$; 在 p 和 q 间添加一条弧, 其权为 p 和 q 共享相

同文件的个数。

3) 在图中加入一个收点,并在该点和图中深度为2的节点间添加弧,每条弧的权都为1。

4) 计算图中种子节点到收点的最大流。

5) 在种子节点的朋友列表中加入由4)得到的节点。

在算法第二步的迭代中,加入到图中的节点可能与种子节点没有任何相同的文档,但与深度为1的节点有相同的文档。尽管种子节点和深度为2的节点当前没有共同的文档,但由于它们都与深度为1的节点有联系,所以很可能不久后它们会至少有一个相同文档。因此,加入这些节点有助于改善社区的效率。每个兴趣社区中的节点数都设置上限,种子节点对社区进行更新时,若社区中的节点数已到达上限,则依据最大利益的原则进行更新,即社区中共享文档数最少的节点被取代。种子节点根据兴趣社区图建立并维护一个朋友节点列表,保存朋友节点的标识、共享相同文档的个数及深度。当节点的共享内容的变化时,根据朋友节点列表通知朋友,并重新衡量与朋友节点的兴趣相似程度。

确定朋友节点后,种子节点保存朋友节点上共享文档的关键词,构造自己的兴趣列表,其主要内容是关键词和相应文档的存储位置。由于结构化P2P系统中每个节点为了共同利益存储其他节点共享文档的关键词,因此节点共享文档的关键词和节点存储的关键词在很大概率下是不同的。种子节点保存的是朋友节点上共享文档的关键词,而不是朋友节点的ID所负责的关键词,增加了种子节点能够定位的资源数量。由于对网络中的每对节点进行兴趣比较代价太大且不现实,为了限制算法的开销,我们限制了每个节点比较的文档数是节点共享资源更新数目的函数。也就是说,只有当本地共享资源变化了一定比例时才更新朋友列表。因此,算法运行频率和开销之间存在折中:如果算法运行频率较高,算法的开销可能会使整个系统的性能降低;另一方面,如果算法很少运行,当节点内容大量改变时,兴趣社区就不能真正体现它当前的兴趣。

进行资源定位时,节点首先从兴趣列表中查找关键词,如果兴趣列表的关键词可以匹配请求,只需再多一跳就可以到达目的节点;如果兴趣列表无法满足搜索请求,节点使用底层Chord的定位算法继续执行搜索定位。

为了得到更好的性能,可以使用缓存加快节点兴趣社区的建立过程。节点离开系统前,将朋友节点的地址存放到本地磁盘中,当节点再次加入系统时,直接与原有的朋友节点联系,比较共享内容,建立兴趣列表。这样,不仅简化了节点的加入过程,也减少了网络带宽的消耗。

3 仿真与性能分析

本文基于MIT的Chord模拟器实现了基于兴趣的资源定位技术的性能模拟,并与原有Chord进行了比较, test1 对应于原Chord算法, test2 对应于基于兴趣局部性的Chord算法。实验参数设置:节点个数、文档个数、请求个数的比例为1:1:10,节点个数分别为100, 500, 1000, 2000, 4000和5000。每个节点的兴趣社区中的节点个数为 $1/2\log N$,即指针个数的一半。由文献[8]的实验结果,节点 n 的第 i 个指针是标识符介于 $n+2^{i-1}$ 和 $n+2^i$ 的前16个节点中距离 n 最近的节点,可使Chord达到近似最佳的延迟。

首先,我们测试了兴趣列表的命中率。命中率是指由兴趣列表直接满足的请求与总请求的比值。如果命中率高,则

兴趣列表有助于资源定位。图3给出了不同网络规模下兴趣列表的平均命中率为56.3%~36.6%。

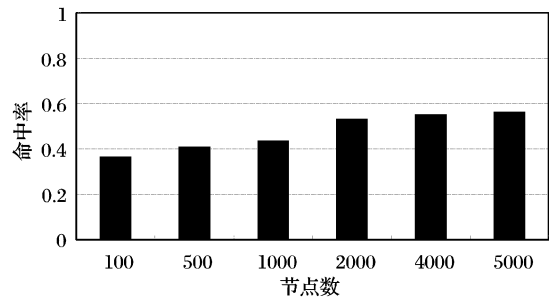


图3 兴趣列表的命中率

实验分别从下面两个测度,对本文提出的技术与原Chord进行比较:

1) 路径长度

路径长度是指请求在到达目标节点时经历的覆盖网络跳数。图4是Chord和基于兴趣的Chord平均查找路径长度的比较。节点 p 发出请求,如果请求不能在本地得到匹配,根据Chord进行查找,其复杂度 $O(\log N)$ 。如果请求目标可以在本地的兴趣列表中得到匹配,则只需再多一跳到达目的节点就可以满足请求,这种情况下满足请求的复杂度是 $O(1)$,极大地缩短了查找路径长度。模拟结果如图4,基于兴趣的Chord的平均路径长度比原Chord缩短了43%~25%。

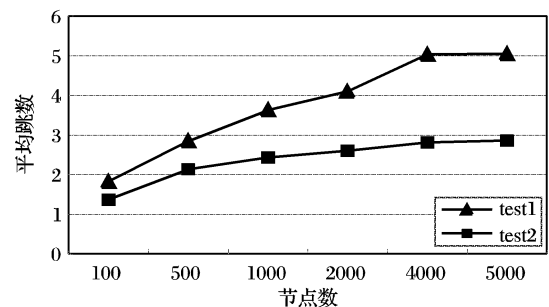


图4 查找路径长度的比较

2) 访问延迟

用户感觉到的延迟减小,主要是由于两个原因:尽可能选择物理网络上最近的节点作为邻居;查找路径长度缩短。两种测试访问延迟的结果如图5,基于兴趣的Chord在访问延迟方面是原Chord的50.6%~38%。

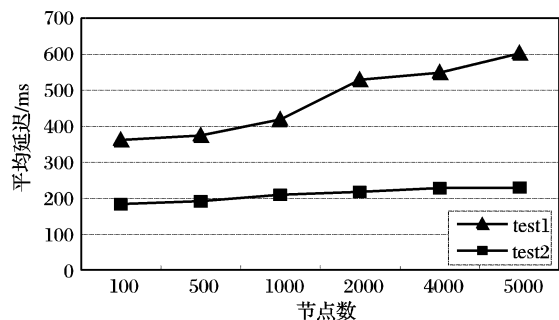


图5 访问延迟的比较

当前客户机的存储能力日益增强,存储朋友节点列表、兴趣列表不会对性能产生影响。并且每个节点的朋友节点数随网络规模指数级增长,因此这种定位技术具有可扩展性,适用于Internet规模的P2P应用。同时,这种技术没有改变Chord的关键词分布方法,因此继承了Chord在负载均衡方面的优点。

(下转第546页)

统溢出率门限对最高接入优先级的话音业务用户没有限制 ($\phi_k = 1$), 所以话音业务用户的呼叫阻塞率明显下降, 系统各类业务用户总呼叫阻塞率略微增加。

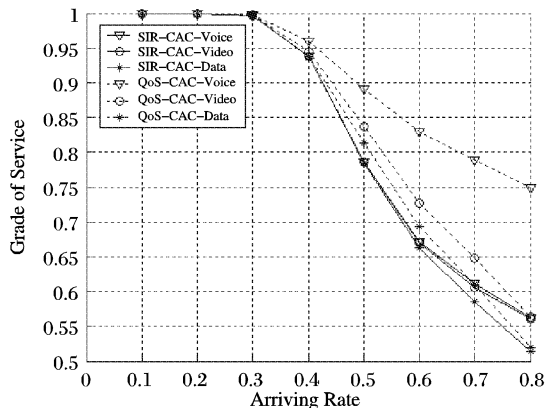


图5 两种 CAC 方案各类业务 GoS

采用了基于 QoS 的 CAC 方案后, 系统溢出率有明显下降, 如图 4 所示, 系统性能得到改善。系统溢出率的下降是以视频、数据业务用户的呼叫阻塞率增加为代价, 当 $p = q = 0.5$ 时, 系统性能综合评价指标如图 5 所示, 话音业务用户 GoS 得到明显提高, 视频、数据业务用户 GoS 也得到了不同程度的改善, 图 6 仿真结果表明, 使用基于 QoS 的 CAC 方案后, 系统 GoS 得到明显地改善。

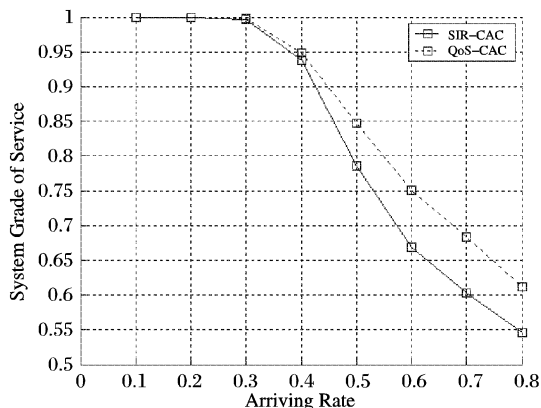


图6 两种 CAC 方案系统 GoS

4 结语

仿真结果可以看出, 本文提出的基于 QoS 的综合业务

CDMA 系统 CAC 方案中 SIR 与系统溢出率联合判决准则在保证系统容量的情况下, 有效地改善了系统性能, 提高了系统的 QoS。同时, 仿真结果还揭示了: 1) 无论采用哪种 CAC 方案, 具有高优先级的业务用户呼叫阻塞率最低, 如话音业务用户, 优先级较低的业务用户呼叫阻塞率较高, 如数据业务用户; 2) 采用基于 QoS 的 CAC 方案可有效抑制系统溢出率, 防止系统性能极为恶劣情况的发生, 系统 GoS 优于单一 SIR 接入判决准则的 CAC 方案, 对无线网络 QoS 的提高具有一定的作用。

值得一提的是, 由于本文没有考虑小区切换用户的情况, 并且为了简化起见, 弱化了数据业务的建模, 所以该 CAC 方案需要进一步完善, 下一步工作将考虑新呼叫用户和切换用户并存情况下的 CAC 方案, 以及结合功率控制等技术的综合 CAC 方案。

参考文献:

- [1] 朱立东, 吴诗其. 多业务无线蜂窝移动通信系统的一种呼叫允许控制策略[J]. 通信学报, 2001, 22(11).
- [2] 关皓, 杜志涛, 李承恕. 综合业务 CDMA 系统的联合呼叫接入-拥塞控制[J]. 电子学报, 1999, 27(11A).
- [3] 李超, 汤汉屏, 郭金淮等. 基于多码 CDMA 系统的综合业务接入控制方案[J]. 电讯技术, 2002, (3): 117-121.
- [4] WHA S J, DONG G J. Call Admission Control for CDMA Mobile Communication Systems Supporting Multimedia Services[J]. IEEE Transactions on wireless communications, 2002, 1(4): 649-659.
- [5] LI F Y, STOL N. A Priority-oriented call admission control paradigm with QoS re-negotiation for multimedia services in UMTS[A]. IEEE VTS 53rd Vehicular Technology Conference[C], 2001, 3. 2021-2025.
- [6] SAMPATH A, SARATHKUMAR P, HOLTZMAN J M. Power control and resource management for a multimedia CDMA wireless system[A]. Proceedings of Personal, Indoor and Mobile Radio Communications[C], 1995. 21-25.
- [7] SO J W. Adaptive traffic prediction based access control in wireless CDMA systems supporting integrated voice/data/video services[J]. IEEE Communications Letters, 2004, 8(12): 703-705.
- [8] SAMPATH A, HOLTZMAN J M. Access control of data in integrated voice/data CDMA systems: benefits and tradeoffs[J]. IEEE Journal on Selected Areas in Communications, 1997, 15(8): 1271-1275.

(上接第 533 页)

参考文献:

- [1] Napster. <http://www.napster.com/index.html> [EB/OL], 2003.
- [2] Gnutella. <http://gnutella.wego.com/> [EB/OL], 2003.
- [3] STOICA I, MORRIS R, KARGER D, et al. Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications[A]. Proceedings ACM Sigcomm 2001[C]. San Diego, CA, USA, 2001. 149-160.
- [4] YANG B, GARCIA-NOLINA H. Efficient Search in Peer-to-Peer Networks[R]. Stanford University, 2001.
- [5] FESSANT FL, HANDURUKANDE S, KERMARREC AM, et al. Clustering in Peer-to-Peer File Sharing Workloads[A]. Proceedings of the 3rd International Workshop on Peer-to-Peer Systems (IPTPS) [C]. San Diego, USA, 2004.
- [6] GUMMADI KP, DUNN RJ, SAROIU S, et al. Measurement, Mod-

- eling, and Analysis of a Peer-to-Peer File-sharing Workload[A]. Proceedings of the 19th ACM Symposium on Operating Systems Principles (SOSP-19) [C]. Bolton Landing, NY, USA, 2003. 314-329.
- [7] SRIPANIDULCHAI K, MAGGS B, ZHANG H. Efficient Content Location using Interest-based Locality in Peer-to-Peer systems[A]. Proceedings of 22nd Annual Joint Conference of the IEEE Computer and Communications Societies [C]. San Francisco, CA, USA, 2003.
- [8] DABEK F, LI J Y, SIT E, et al. Designing a DHT for Low Latency and High Throughput[A]. Proceedings of the 1st Symposium on Networked Systems Design and Implementation (NSDI '04) [C]. Berkeley, CA, USA, 2003. 85-98.