

# 一种改进的基于键聚类的 P2P 网络搜索算法

关 中

(广州城市职业学院 广州 510230)

**摘 要** P2P 搜索问题已成为目前学术界的研究热点,Key clustering 算法将路由空间分成 HUB 和 AUT 两层,从全局角度进行有序搜索,借鉴 Small-world 领域的研究成果,在路由表中以一定概率插入连接远距离节点的快捷连接,以缩短平均路径长度。仿真试验表明,引入快捷连接的 Key clustering 算法具有良好的搜索能力、扩展性和容错能力。

**关键词** 搜索算法,索引表,P2P,分布式路由

## An Improved Key Clustering Searching Algorithm in P2P Network

GUAN Zhong

(Guangzhou City Polytechnic, Guangzhou510230)

**Abstract** An efficient search technique has become a crucial research problem in P2P system. Key clustering algorithm divides routing space into AUT layer and HUB layer. Such an algorithm can do a well-ordered search from a global view. To improve scalability, a few shortcuts with distant peers are inserted into routing tables with some probabilities, and average path length is reduced. Result shows key clustering algorithm with shortcuts is efficient and scalable.

**Keywords** Searching algorithm, Indexing table, Peer-to-Peer, Distributed routing

## 1 引言

近年来,对 Peer-to-Peer(P2P)数据存储及共享系统的研究已经成为互联网领域研究的焦点之一。其中搜索算法是决定 P2P 系统性能的重要因素,而搜索算法的性能主要取决于查询消息所采用的路由策略。目前已有的路由策略主要分为基于信息路由和盲目消息路由两类。在基于信息的路由策略中,依据自身已有的邻居结点信息和资源信息,有区别地选择邻居结点转发查询,以提高发现目的资源的效率。在盲目路由方式中,采用广播或者随机选取部分结点进行转发的策略。盲目消息路由算法简单、有效,但是网络开销太大,为此提出了许多算法,以减少广播算法的缺点。例如,在 Random walks 算法中设置  $k$  个 walker,walker 在网络中漫游遍历而不是扩散查询消息。

本文设计了一种基于键的聚类消息路由算法(Key Clustering),通过逐步调整将节点索引记录按照一定中心进行聚类,在物理网络层上建立具有全局视角的路由层。为了减少平均路径长度,利用小世界(small-world)的研究成果,将确定式聚类改进为带有随机性的概率式聚类,在导航表中引入指向远距离节点的快捷连接。

## 2 Key clustering 算法

### 1) 路由表构成

Key Clustering 算法(简称 KC 算法)的索引采用分布式哈希表(Distributed Hashing Tables,简称 DHT)<sup>[1]</sup>,利用哈希函数将文件的属性(如文件名或描述性关键词)映射为键(key)。将 key 按照升序排列,并首尾相接,形成环状空间, key 空间是位于物理节点层之上的上层结构(overlay),具有可搜索性。搜索时,query 也映射为 key。定义 key 之间的距

离为它们在环状 key 空间上的最短距离,即  $D(a, b) = \min\{|a - b|, |M - a - b|\}$ ,其中  $M$  为空间中的 key 的总数。

表 1 KC 算法的路由表

Node 002 center = 4350					
HUB	Key	Node ID	AUT	Key	Node ID
	3210	210		4218	260
	3850	80		4232	400
	...			...	

每个节点拥有一个由中央服务器分配的聚类中心(center),节点根据搜索表现更新路由表内容,使内容向着 center 进行聚类。各个节点的 center 在 key 空间中均匀分布。为了保证系统能够及时追踪时刻变化的查询热点,中央服务器会根据近期 query 的分布为节点分配 center,以保证整个系统的索引均衡性。

节点在本地需要维护一张长度为  $K$  的路由表(见表 1)。路由表分成导航表(HUB)和主题表(AUT)两部分,长度分别为  $K_H$  和  $K_A$  ( $K_H + K_A = K$ )。AUT 表记录节点 ID 和代表其文件特征的 key, HUB 表记录节点 ID 及其 center。从功能上看, AUT 层面从全局角度按 key 的数值顺序重新编排文件记录, HUB 层面则按节点 center 的数值顺序排列节点,形成有序的路由层。系统运行时,二者分别按照一定聚类策略进行更新。

### 2) 搜索过程

搜索时首先检索本地文件,若与 query 匹配,则沿原路回送结果,同时更新路径的节点路由表;否则在 AUT 表中查找 query,若匹配,则前送消息至合适的节点;若 AUT 表不存在匹配节点,从 HUB 表中寻找最接近 query 的节点(即 center 与 query 接近),并发送 query 至该节点。

关 中 讲师,研究方向:网络系统、数据库系统、嵌入式系统、软件工程等。

### 3) 路由表更新策略

设搜索路径的长度为  $L$ , 搜索命令为  $q$ , 从起始节点到目标节点依次是:  $n_1, n_2, \dots, n_L$ 。由于路径上的节点参与了对  $q$  的搜索, 故可以认为它们与  $q$  的相关性较大, 因此, 在搜索成功后, KC 算法会对这些节点的路由表进行更新操作。

#### HUB 表更新策略

对于  $n_x (x=1, 2, \dots, L-2)$ , 进行 HUB 表的更新:

① 令等待加入的新记录  $key_{ins} = center(n_{L-1})$ ,  $center(n_{L-1})$  即为  $n_{L-1}$  的聚类中心, 相应的记录为  $\langle key_{ins}, n_{L-1} \rangle$ 。之所以选择  $n_{L-1}$  为学习目标, 是因为  $n_{L-1}$  作为路径上最后一个消息传递者, 为搜索成功做出了最重要的贡献;

② 若 HUB 表未滿, 则加入新记录  $\langle key_{ins}, n_{L-1} \rangle$ , 更新结束; 若 HUB 表已滿, 则执行③;

③ 待删除的旧记录  $key_{del}$  为  $n_x$  的 HUB 表中距离  $center(n_x)$  最大的记录

$key_{del} = \text{Max} \{ D(key_x, center(n_x)) \}$ ,  $key_x$  是  $n_x$  的 HUB 表中的任意一条记录;

④ 如果  $D(key_{ins}, center(n_x)) < D(key_{del}, center(n_x))$ , 则用新记录取代旧记录; 否则不做任何改动。

#### AUT 表更新策略

对于  $n_x (x=1, 2, \dots, L-1)$ , 进行 AUT 表的更新:

① 因为在  $n_L$  处搜索到  $q$ , 所以  $n_L$  的文件中必包含  $q$ , 路径上的节点将记录该信息。令等待加入的新记录  $key_{ins} = q$ , 相应的记录为  $\langle key_{ins}, n_L \rangle$ ;

若 AUT 表未滿, 则加入新记录  $\langle key_{ins}, n_{L-1} \rangle$ , 更新结束; 若 AUT 表已滿, 则执行③;

② 待删除的旧记录  $key_{del}$  为  $n_x$  的 AUT 表中距离  $center(n_x)$  最大的  $key$ , 即:  $key_{del} = \text{Max} \{ D(key_x, center(n_x)) \}$ ,  $key_x$  是  $n_x$  的 AUT 表中的任意一条记录;

③ 如果  $D(key_{ins}, center(n_x)) < D(key_{del}, center(n_x))$ , 则用新记录取代旧记录; 否则, 不做任何改动。

经过足够多次的更新后, HUB 和 AUT 中的  $key$  可以有效地聚类于  $center$ 。

## 3 KC 算法的改进

基于 Klenberg 的研究成果, 对 KC 算法的 HUB 表更新策略进行改进, 提出 EKC (Enhanced Key Clustering) 算法, AUT 表更新策略仍与 KC 算法相同。

EKC 算法的 HUB 表更新策略如下:

对  $n_x (x=1, 2, \dots, L-2)$ , 进行 HUB 表的更新:

① 令等待加入的新记录  $key_{ins} = center(n_{L-1})$ ,  $center(n_{L-1})$  即为  $n_{L-1}$  的聚类中心, 相应的记录为  $\langle key_{ins}, n_{L-1} \rangle$ ;

② 若 HUB 表未滿, 则加入新记录  $\langle key_{ins}, n_{L-1} \rangle$ , 更新结束; 若 HUB 表已滿, 则执行③;

③ 计算 HUB 表中每条记录的删除概率, 记  $u = center(n_x)$ , 则  $key v$  的删除概率为:

$$P(v) = \frac{D(u, v)^{-1}}{\sum_w D(u, w)^{-1}}$$

其中  $w$  为 HUB 表中的任意  $key$ 。根据删除概率随机选择一个  $key$  为待删除记录, 记为  $key_{del}$ ;

④ 重新计算  $key_{del}$  和  $key_{ins}$  的删除概率:

$$P(key_{del}) = \frac{D(u, key_{del})^{-1}}{D(u, key_{del})^{-1} + D(u, key_{ins})^{-1}}$$

$$P(key_{ins}) = \frac{D(u, key_{ins})^{-1}}{D(u, key_{del})^{-1} + D(u, key_{ins})^{-1}}$$

根据删除概率随机确定删除对象, 对 HUB 表做出相应调整。EKC 算法按照概率更新 HUB 表, 可以为 HUB 表引入少数的 shortcut。

## 4 改进算法的性能分析与比较

### 4.1 实验环境

本文使用均匀随机图模型作为网络拓扑, 节点维护的邻居数 (节点度) 在最小值到最大值之间均匀分布, 平均邻居节点度为 3.5。整个网络包括 10000 个节点。每次搜索时, 随机选定 query 发起节点, 随机选定 query。在搜索成功后更新搜索路径节点的路由表。

### 4.2 两种算法的平均路径长度比较

图 1 比较 KC 和 EKC 在不同节点数量下的平均路径长度变化, 可以看出, 随着节点数增加, KC 曲线的抬升速度大大快于 EKC 曲线, 与理论上的指数趋势相符。EKC 受节点数量增长的影响小, 比 KC 具有更好的推广性。在性能上有进一步的提高。由于 EKC 的 HUB 存在少量的 shortcut, 加快了搜索进程, 使之平均路径长度比 KC 降低 37%, 成功率提升 13%。

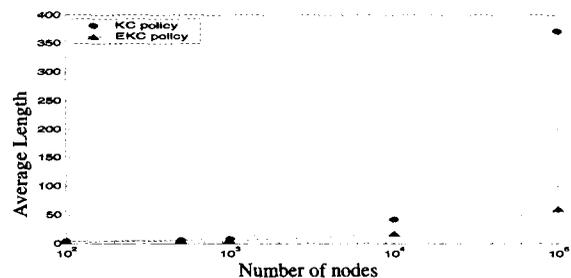


图 1 不同节点数量下的平均路径长度

### 4.3 两种算法的 HUB table 表 key 分布

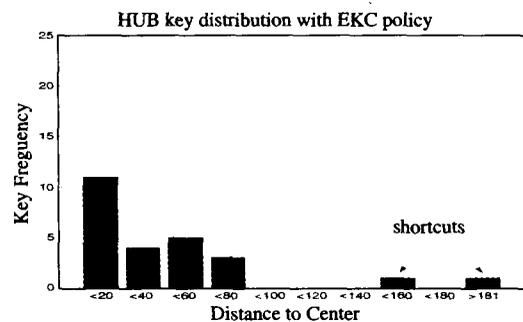


图 2 HUB table Key of KC

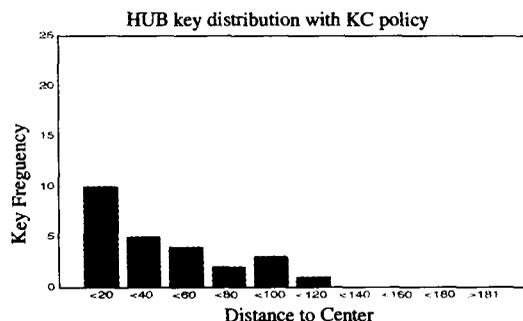


图 3 HUB table Key of EKC

星的覆盖域把 MEO 卫星分组以实现层次性链路状态更新,由 MEO 卫星汇总覆盖域内所有 MEO 卫星的链路状态通告,但引入的 MEO 层卫星不仅增加了网络中的卫星数目,而且在 MEO 层星座仍需广播汇总后的 MEO 层链路状态通告,从而使其总通信开销远大于 DRRA 算法的星间通信开销。而对于 MLSR,采用 GEO、MEO、LEO 三层星座的组网结构,网络中的卫星总数最多,组网结构也最复杂,尽管各层星座路由信息的计算均由上一层星座完成,LEO 星座的路由信息的计算在 MEO 星座细化完成,但仍需 GEO 星座进行全网通信节点及其邻接链路状态信息的收集,因此 MEO 星座处理最复杂,GEO 与 MEO 星座之间星际链路  $IOL_{GEO \rightarrow MEO}$  上的辅助通信负载最重,即 MLSR 算法的星间通信开销最大。而抗毁路由算法 DRRA 的通信资源开销主要包括群内成员状态收集与路由信息的分发、星座间群首与群管理者之间的状态与路由信息的传递、以及各路由表计算者之间  $RTC(x)$  的状态与路由信息的通告交换,相对 SGRP 和 MLSR 两种路由算法,抗毁路由算法 DRRA 的通信资源开销是最小的。

**总结** 本文提出了一种新的基于 GEO/MEO 星层组网的抗毁路由算法 DRRA 具有计算复杂度相对较低、存储容量相对较小、支持卫星节点失效运行的特点和功能。这主要基于:①卫星网络中通信节点间链路冗余设计方案;②卫星网络中通信节点失效时间间隔失效节点视距范围内次相邻节点链路修补方案;③MEO 星座分群群首失效时采用群首选择备份机制;④卫星网络中分群群管理者失效时采用群首兼容群管理者功能的角色扩充机制。另外较其它相关算法而言,提出的路由算法在计算/存储复杂度以及通信资源占用率和数据包路由最短延时等方面具有优势,这主要是提出:①基于 GEO 通信覆盖域的 MEO 星座分群管理机制;②通过分群来实现

分布与集中相结合的路由更新策略;③基于状态更新的信息增量修正 IIM 路由计算方法;④卫星网络中通信节点间链路冗余设计方案。

### 参考文献

- 1 Werner M, et al. ATM-Based Routing in LEO/MEO Satellite Networks with Intersatellite Links. IEEE Journal on Selected Areas in Communications, 1997, 15(1)
- 2 Werner M. A Dynamic Routing Concept for ATM-Based Satellite Personal Communication Networks. IEEE Journal on Selected Areas in Communications, 1997, 15(8)
- 3 Chang H S, et al. Topological Design and Routing for Low-Earth Orbit Satellite Networks. In: Proc. of IEEE GLOBECOM, 1995. 529~535
- 4 Gounder V V, Prakash R, Abu-Anlra H. Routing in LEO-based satellite networks. In: Proc. of IEEE Emerging Technologies Symp. Wireless Communications and Systems, Apr 1999
- 5 Chang H S, Kim B W, Lee C G, et al. FSA-based link assignment and routing in low-Earth orbit satellite networks. IEEE Transactions on Vehicular Technology, 1998, 47(3): 1037~1048
- 6 Werner W, Berndt G, Edmaier B. Performance of optimized routing in LEO intersatellite link networks. In: Proceedings of IEEE 47th Vehicular Technology Conference, vol 1, May 1997. 246~250
- 7 Ekici E, Akyildiz I F, Bender M D. A distributed routing algorithm for datagram traffic in LEO satellite networks. IEEE/ACM Transaction on Networking, 2001, 9(2): 137~147
- 8 白建军. 天基信息系统组网关键技术研究:[学位论文]. 国防科技大学, 2005
- 9 易先清, 赵阳, 罗雪山. 一种基于 GEO MEO 星层组网的结构组织与数据传输分析, 2006
- 10 Lo M W. Satellite constellation design. IEEE Computing in Science and Engineering, January/February 1999. 58~67
- 11 李建东. 信息网络理论基础. 西安电子科技大学出版社, 2001
- 12 Chen C, Ekici E. A routing protocol for hierarchical LEO/MEO satellite IP networks. ACM/Kluwer Wireless Networks (WINET) Journal, July 2005. 507~521
- 13 Chen C, Ekici E, Akyildiz I F. Satellite Grouping and Routing Protocol for LEO/MEO Satellite IP Networks. In: Proc. of the Fifth International ACM Workshop on Wireless Mobile Multimedia
- 14 Akyildiz I F, Ekici E, Bender M D. MLSR: A novel routing algorithm for multi-layered satellite IP networks. IEEE/ACM Transaction on Networking, 2002, 10(3): 411~424

(上接第 73 页)

图 2, 图 3 给出了两种算法的 HUB table 表 key 分布, key 与 center 的距离划分为若干区间段, 统计每段 key 数量, 考察算法对 key 分布的影响。EKC 的 HUB 调整算法在考虑与 center 距离基础上加入随机性因素, 以一定概率决定新记录是否加入, 不同于 KC 的严格聚类作法。因此, EKCHUB 中存在少数非聚类 key。事实上, 聚类 key 起到了近邻连接的作用, 非聚类 key 相当于 shortcut。

#### 4.4 搜索算法成功率

节点的自由退出是无组织系统必须要考虑的问题。图 4 显示了节点退出对 EKC 算法的成功率的影响, 以所有节点在线的成功率为基准, 纵轴代表当前成功率与基准的比值。试验中, 要保证系统中存在 3 份以上的满足 query 的资源, 以保证在部分节点退出时, 搜索仍可能找到结果, 图 4 中, 随着节点退出比例的增大, 曲线开始下降缓慢, 当退出率=35%时, 成功率仍能保持在 80%以上。之后, 曲线才加速下降, 当退出率=50%时, 成功率仅为原来的 10%左右。

在一定的退出率下, 除了退出节点的路由表不再发挥作用, 工作节点的路由表中涉及到退出节点的记录也失效, 图 4 说明尚存的少数 HUB 连接仍能有效的指引搜索方向, 同时, 聚类 AUT 表缩小了目标查找范围, 节省了消息发送数量, 可见 EKC 具有良好的容错性。相比之下, 组织无序 KC 算法搜索能力依赖于有效记录数量, 节点退出时, 搜索的盲目性更加突出。

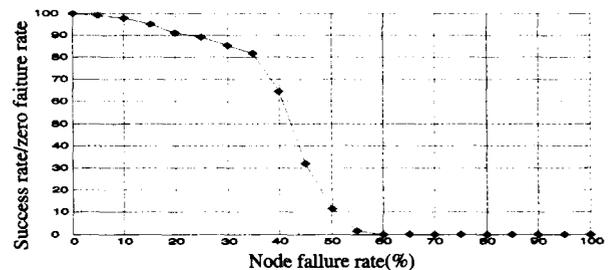


图 4 搜索算法成功率

**结论** 针对无组织 P2P 系统的分布式路由算法搜索无序问题进行研究, 提出了 Key clustering 算法, 它将路由空间分成 HUB 和 AUT 两层, 从全局角度构建路由表, 从而实现有序搜索。借鉴小世界的研究成果, 在路由表中以一定概率插入连接远距离节点的快捷连接, 在理论上, 平均路径长度的期望值改善为  $O(\log^2 N)$ , 提高了算法的可扩展性。仿真试验表明, 引入快捷连接的 Key clustering 算法具有良好的搜索能力、可扩展性和容错能力。

### 参考文献

- 1 Sam J. Semantically Routing Queries in Peer-to-Peer Networks. Proceedings of the International Workshop on Peer-to-Peer Computing, 2002, 18(2): 152~160
- 2 Balakrishnan. Looking up data in p2p systems. Communications of the ACM, 2003, 46(2): 43~48
- 3 Watts D H. Collective dynamics of small-world networks. Nature, 1998, 393: 440~442
- 4 Kleinberg J. The small-world phenomenon: an algorithmic perspective. Cornell Computer Science Technical Report, 99-1776, 2000, 32(1): 108~126