基于活跃节点的KAD网络高效查询方法

严 鹤 刘 威 张 戈 程文青
(华中科技大学电子与信息工程系湖北省智能互联网技术重点实验室 武汉 430074)

摘 要 KAD网络的查询性能受到P2P节点动态特性的影响。以节点ID在路由表中重复出现的次数作为节点活跃度的表征,提出了一种基于活跃节点的KAD网络查询方法。该方法首先通过网络爬虫获取KAD网络的分布式路由信息,然后获取当前活跃节点的集合,最后将活跃节点作为查询过程中的备选节点。实验结果表明,与现有的查询方法相比,该方法在查询时间上减少了60%,在获取文件范围上提高了18%,较显著地提升了查询的效率。

关键词 对等网络,KAD,活跃节点,查询性能
中图法分类号 TP393 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2014.12.013

Efficient Lookup Method Based on Highly Available Peers in KAD

YAN He LIU Wei ZHANG Ge CHENG Wen-qing
(Internet Technology and Engineering R&D Center, Department of Electronics and Information Engineering, Huazhong University of Science and Technology, Wuhan 430074, China)

Abstract The lookup performance in KAD is affected by the dynamics of peer participation. By studying the KAD routing mechanism, we found the times of ID appearance in routing tables can be the measurement of the peer availability. Based on the highly available peers, we proposed a new lookup method for KAD peers. At first, the KAD routing information is collected by a crawler. Then the highly available peers are selected as the lookup candidates. Experiment results show that, compared with the existed lookup scheme, our method can reduce 60% of average lookup time and obtain 18% more files in lookup results.

Keywords Peer-to-Peer network, KAD, Highly available peers, Lookup performance

1 引言

KAD网络是一个拥有几百万用户的大型Peer-to-Peer(P2P)网络,其查询性能一直为学术界所关注。KAD网络基于Kademlia协议,其节点和资源的定位依靠KAD路由的迭代查询来完成。由于网络中节点动态地加入或退出,使得系统路由表中的部分节点离线或不可连接,进而影响整个网络的查询性能[1]。有测量结果表明KAD网络的路由表中32%的节点是不可连接的,如果在查询过程中遇到这些失效节点[2],则会严重影响查询性能。

KAD网络中也有部分节点始终在线,后文将该类节点称为活跃节点。目前已经出现了一些对KAD网络节点的活跃度的研究。在文献[3]中,将节点的活跃度定义为节点数据可被访问的概率。文献[4]依据节点的活跃度将节点分为两类:始终在线的节点、循环地接入和退出网络的节点;并通过对节点的两种行为进行组合建模,分析节点的活跃度。文献[5]通过长期监听节点在线和离线的行为,测量得出活跃节点占所有KAD节点的0.6%。文献[6,7]指出了KAD节点ID的重复、UDP端口变化和IP地址变化对节点活跃度的影响。这些研究虽然观测到活跃节点的一些特性,却没有给出一种提取KAD网络中活跃节点的方法,也无法利用活跃节点长期在线的特性来提高KAD网络的查询性能。

本文通过对KAD路由的研究,发现节点在线时间越长,其对应的KAD节点ID出现在其它节点路由器中的概率越大。因此,本文提出用KAD节点ID在其它节点路由表中重复出现的次数表征节点的活跃度,并以活跃节点作为查询备选节点的快速查询方法。实验结果表明,该方法在降低KAD网络平均查询时间的同时,可以获得更多的查询文件结果。

本文第2节对KAD网络查询性能进行建模分析,给出基于活跃节点的查询方法的研究动机;第3节给出了查询方法的实现过程;第4节对现有的KAD网络查询过程进行改进,并与现有的KAD网络查询过程进行性能对比分析;最后对全文进行总结。

2 KAD网络查询性能的研究

2.1 KAD网络中查询性能的模型

文献[2]指出,KAD网络查询性能影响因素包括无效节点的概率P_{sale}、两节点间信息交换的延迟d、到达目的节点的

到稿日期:2014-01-23 返修日期:2014-03-24 本文受国家自然科学基金资助项目(61301127,61371080)资助。
严 鹤(1982-),男,博士生,主要研究方向为P2P网络、网络测量,E-mail:isaacyh@gmail.com;刘 威(1977-),男,博士,副教授,CCF会员,主要研究方向为内容中心网;张 戈(1983-),男,博士生,主要研究方向为网络测量;程文青(1964-),女,教授,博士生导师,主要研究方向为高速通信网络。

跳数 h、发送路由请求的节点个数 α、路由请求返回的节点个数 β 和节点等待应答信息的时间 t。

KAD 网络查询过程可以使用这些因素进行刻画。假设第一跳发送 α 个路由请求，在 α 个请求中只收到 $\alpha(1-P_{stale})$ 个应答。针对每个应答，$\gamma=\min(\alpha,\beta)$ 个邻居节点被返回。第二跳发送路由请求的最大节点个数 $\rho_{2,\max}=\alpha(1-P_{stale})\gamma$。在接下来的一跳中，只有连接节点中的 $(1-P_{stale})$ 应答消息，每个应答节点返回 γ 个邻居节点。则在第 i 跳时发送请求节点的最大个数 $\rho_{i,\max}$ 可以表示为：

$$\rho_{i,\max}=\alpha[(1-P_{stale})\gamma]^i \tag{1}$$

上述结果可以扩展到 h 跳时的情形，发送消息的累计最大值 $\rho_{h,\max}$ 为：

$$\rho_{h,\max}=\alpha\sum_{i=0}^{h-1}[(1-P_{stale})\gamma]^i \tag{2}$$

发送请求节点的最大个数 $\rho_{i,\max}$ 越大，表示查询消息请求成功的概率越大。式(2)给出了各因素与查询结果的关系。

2.2 KAD 网络查询性能的影响因素分析

在各种因素中，节点交换延迟 d 和目的节点跳数 h 由 KAD 系统的实时情况决定，而发送请求个数 α、返回节点个数 β 和等待时间 t 可以通过协议参数修改来调整，所以文献[2]通过改变 α、β 和 t 的值来提高查询性能。例如，文献[2]通过实验表明，将 KAD 网络等待时间 t 从缺省值 3 秒减少到 0.5 秒，可以带来查询速度 60%左右的提升，但是返回结果的文件数量也大幅减少。作者认为，高效率的 KAD 查询是指在最短的时间内返回最多的查询结果。在查询协议的操作参数中，α、β 影响的是查询范围，参数 t 影响的是查询速度，片面地调整一个方面并不能实际提升查询的效率。

由式(2)可知，无效节点的概率 P_{stale} 也是影响查询性能的重要参数。$(1-P_{stale})$ 刻画了目前查询集合对 KAD 网络的认知水平，其值越大，查询消息请求成功的概率越大。与查询协议的操作参数 α、β 和 t 不同，$(1-P_{stale})$ 直接提升查询的有效性，是影响查询效率的本质因素。本文提出，在查询过程中把无效节点替换成活跃节点，将提升 $(1-P_{\text{stale}})$，进而可以提高 KAD 网络的查询性能。

3 基于活跃节点的 KAD 网络查询方法

获取活跃节点的主要依据是 KAD 网络的路由表信息，以下进行分析。

3.1 KAD 网络路由更新机制分析

在 KAD 网络中，节点路由表由通过多个称之为 K 桶的列表构造而得。以 eMule 系统为例，每个 K 桶维持一个〈IP 地址，UDP 端口，KAD ID〉列表，记录不多于 k 个距离节点 2^i 和 2^{i+1} $(0\leqslant i<128)$ 的节点的信息。k 是为平衡系统性能和网络负载而设置的一个常数，在 eMule 中 $k=10$。两个节点的距离则是基于位异或的二进制运算来进行计算，节点 x，y 的距离为 $d(x,y)=x\oplus y$，该距离称为 XOR 距离。

每个 K 桶列表内的节点信息是按照节点与其进行消息通信的先后顺序排列的。最先进行交互的节点放在列表头部，最后交互的节点放在列表尾部。每当 KAD 消息被接收时，发送者的节点 ID 就被用来更新对应的 K 桶。如果发送者的节点 ID 已经存在于这个 K 桶中，则把对应项移到该 K 桶的尾部。如果发送者的节点 ID 没有在这个 K 桶中，且该 K 桶的记录项小于 k 个，则直接把发送节点的〈IP 地址，UDP 端口，KAD ID〉信息插入队列尾部。如果目前 K 桶的记录已满，即已有 k 个记录，则探寻头部的第一个节点 z。如果 z 没有响应，则从 K 桶中移除 z 的信息，并把发送节点的〈IP 地址，UDP 端口，KAD ID〉信息插入队列尾部；如果 z 有响应，则把 z 的信息移至队列尾部，同时忽略发送节点的信息[8]。

由 K 桶的更新机制我们可以看到，如果一个节点一直在线，它的节点 ID 会一直保留在与其交互的节点路由表中，不会从这些节点的 K 桶中被剔除。于是我们推断，由于节点间不断的消息交互，节点在线时间越长，其节点 ID 出现在其它节点路由表中的概率越大。因此本文用节点 ID 在其他节点路由表中出现的频率作为依据来获取活跃节点。

3.2 节点路由信息的爬取和活跃节点的标记

由于需要获取节点 ID 在其他节点路由表中出现的频率，我们设计并实现了 KAD 网络中的节点路由信息爬虫。策略如下：首先爬虫向种子节点发送路由请求，获取并记录新的节点。以这些新的节点为基础采用广度优先搜索算法进行迭代路由查询，通过已知的节点信息发现新的节点。当节点信息被路由请求响应返回时，检测该节点的 ID 是否已知。如果已知，则 ID 的出现次数增加 1，如果未知，则将以新节点的形式保持 ID 信息。最后，将爬虫记录的 ID 出现次数排序，剔除掉对 KAD 消息不作应答的“沉默”节点，ID 排名靠前的节点即标记为活跃节点。爬取的全网的路由信息结果将被写到基于文档的 Mongo 数据库中保存。

3.3 基于活跃节点的 KAD 网络查询

KAD 网络中的查询过程是由查询节点发起的。网络中的邻居节点应答后，将更近的节点信息返回给查询节点。查询节点选择返回信息中距离目标 XOR 更近的邻居节点再次发送路由请求，收到更近的邻居节点信息。如此迭代，查询节点最终确定一个离目标 XOR 较近的节点列表。最后查询节点对列表中的节点进行内容询问，获得对应的内容信息。

利用活跃节点的较长时间在线的特性，本文对 KAD 网络查询方案进行了改进，具体实现如下：

1)用爬虫获取 KAD 网络中的活跃节点，每天一次，选择 ID 重复次数排名前 10000 的节点作为活跃节点候选集。利用设计的爬虫执行该操作耗时约 10 分钟。

2)对于查询节点迭代路由查询过程中的每一跳，记录在节点等待应答信息的时间 t 内没有响应的节点 $Peer_{stale}$。

3)查询节点将活跃节点候选集中的节点与 $Peer_{stale}$ 进行 XOR 距离计算，得出距离最近的活跃节点 $Peer_{active}$。

4)查询节点用 $Peer_{active}$ 替换路由中的 $Peer_{stale}$，重新发送该跳的查询请求，以获取更多的邻居节点路由应答信息。

由于 KAD 网络中节点 ID 产生的 Hash 算法是均匀分布的，因此选取的这 10000 个节点也均匀分布在整个 Hash 空间。对于 KAD 不同子空间的查询，都有合适的活跃节点对失效节点进行替换。

4 查询方案的性能评估

我们将本文提出的查询方案称为“活跃查询”，分别从节点活跃度、查询速度和查询范围 3 个方面对活跃查询进行评估。

4.1 节点活跃度评测

从 2013 年 7 月 23 日到 2013 年 8 月 12 日，我们连续爬

取 KAD 网络中的活跃节点，期间选择了 600 个随机普通节点和 600 个爬取到的活跃节点作为观察对象，每隔 5 分钟检测节点是否在线，获得的实验结果如图 1 所示。

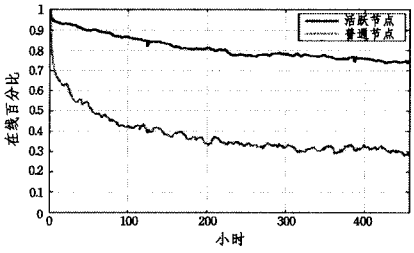

图 1　活跃节点与普通节点对比

由图 1 可以看出，活跃节点的在线比例始终高于普通节点。随着时间的推移，两组节点的在线比例逐渐下降，前 100 个小时内普通节点的下降速度较快，其后下降趋于平缓。这个现象表明，普通节点中短期参与 KAD 网络的节点会在 100 小时内逐渐退出网络，其后剩余节点多为稳定节点。而获取的活跃节点在线比例的下降趋势一直较为平缓，即使 20 天(480 小时)后也仍高于 74%。上述结果表明，本文获取的节点的活跃度较高，可以作为后续查询的基础。

文献[5]定义的稳定节点是指在线平均时间长、离线时间短的节点。很明显可以看出，我们获取的活跃节点活跃度高、在线时间长，符合文献[5]提出的稳定节点的定义。而从这些活跃节点的在线特征也可以看出，文献[5]中定义的稳定节点活跃度较高。

4.2　查询速度的评估

查询速度的性能以获取到指定数量文件的平均耗时来评估。我们分别用 100 个活跃节点和 100 个普通节点作为查询开始时的入口节点。随机选择 100 个非热门的关键字，针对每个关键字分别对活跃节点和普通节点构成的查询入口节点进行关键字查询测量。测量系统将记录每次活跃查询与普通查询得到相同数量文件所需要的时间。对这 100 次查询测量结果取均值，如表 1 所列。

表 1　查询平均耗时的比较(单位：ms)

查询类型	获取 10 个文件耗时	获取 50 个文件耗时	获取 100 个文件耗时	获取 300 个文件耗时
普通查询	2360	10594	10922	11281
活跃查询	969	1172	1531	4438

从表 1 可以看出，查询相同数量的关键字文件，活跃查询所需时间远小于普通查询。以 eMule 设置的缺省查询返回值 300 个文件为例，活跃查询比普通查询在时间上提升了 60.66%。

4.3　查询范围的评估

对相同关键字的不同查询获得的文件数量是不一样的。查询范围的性能通过获取到的文件数量来进行评估。我们从 817 部电影和 189 部电视剧的名称中提取了 1200 个关键字，这 1200 个关键字的哈希值均匀分布于整个哈希区间，以保证这些关键字的查询能覆盖整个网络。测量系统同时用普通查询和活跃查询搜索这些关键字，其中邻居节点个数设置为 40。测量系统记录每次普通查询和活跃查询路由迭代的跳数总和、应答节点个数和最终查询到的文件数目。对这 1200 个关键字的查询测量完成后，将测量结果取均值后的结果如表 2 所列。

表 2　查询范围能力比较

查询类型	平均文件数量(个)	平均应答数量(个)	平均总跳数(个)
普通查询	1357.25	142.23	680.12
活跃查询	1609.52	155.57	851.75

从表 2 可以看出，相比普通查询，活跃查询查到的文件个数平均多 18.57%，且应答节点个数以及跳数总和也较正常查询过程多。

结束语　KAD 网络的查询性能一直为学术界关注。现有工作主要着眼于查询参数的修改。文献[2]通过修改协议参数 t 获得约 60%的速度性能的提升，但其却是以减少返回文件数量为代价的。与前期工作的思路不同，本文的工作以提升查询集合的节点活跃度为思路，可以提升查询的整体效率。

本文提出利用路由表中节点 ID 重复出现的次数表征节点的活跃度。我们依据上述原理用爬虫快速获取了 KAD 网络中的活跃节点，利用活跃节点代替查询过程中失效的普通节点。与现有的随机查询方法相比，该方法在查询时间上减少了 60%，在获取文件范围上扩大了 18%。本文在 KAD 查询中引入了额外的爬取过程。本文设计的爬虫完成一次全网爬取耗时约 10 分钟，而根据文献[6]的报道，目前较快的 KAD 爬虫爬取全网只需 8 分钟。我们认为爬取环节的时间开销与最终性能的提升相比是值得的，因为从整体上看，本文的方法已显著提高了 KAD 网络查询效率。

参 考 文 献

[1]　刘丹，谢文君. 一种面向 P2P 空间查询的路由恢复方法[J]. 计算机科学，2013，39(12)：47-50

[2]　Steiner M，Carra D，Biersack E. Faster content access in KAD [C]// IEEE P2P. 2008：195-204

[3]　Xin Q，Schwarz T，Miller E L. Availability in global peer-to-peer storage systems[C]// International Workshop on Distributed Data & Structure (WDAS). 2004

[4]　Douceur J R. Is remote host availability governed by a universal law? [J]. ACM SIGMETRICS Performance Evaluation Review，2003，31(3)：25-29

[5]　Zhang Y，Xiao C，Zhang H. Analysis of session sequences of stable peers in P2P systems[C]// 2013 3rd IEEE International Conference on Broadband Network and Multimedia Technology (IC-BNMT). 2010：1005-1009

[6]　Steiner M，En-Najjary T，Biersack E W. Long term study of peer behavior in the KAD DHT [J]. IEEE/ACM Transactions on Networking，2009，17(5)：1371-1384

[7]　Yu J，Fang C，Xu J，et al. ID repetition in Kad[C]// IEEE P2P. 2009：111-120

[8]　Maymounkov P，Mazières D. Kademlia：A peer-to-peer information system based on the XOR metric[C]// First International Workshop on Peer-to-Peer Systems (IPTPS). 2002：53-65