DHT负载均衡的必要性

聂晓文 卢显良 李 梁 徐海湄 蒲 汛
(电子科技大学计算机学院 成都 610054)

摘 要 在分布式哈希表(DHT)中,节点ID通常随机选择,但这并不意味着DHT是负载均衡的。仿真结果表明,Chord网络中的负载是不均衡的。在总结对该问题相关工作的基础上,理论上证明了DHT算法本质上的不均衡性,并精确给出节点负载不均衡性的上界范围,仿真验证了分析结论。

关键词 对等网,分布式哈希表(DHT),负载均衡,概率分布

中图法分类号 TP393 **文献标识码** A

On the Necessity of Load Balance in DHT

NIE Xiao-wen LU Xian-liang LI Liang XU Hai-mei PU Xun
(School of Computer Science & Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China)

Abstract In the distributed hash table (DHT), the identifiers of nodes are chosen at random, but this does not mean that the DHT is load-balanced. The simulation in Chord has shown that the load is imbalanced. The paper summarized the previous work on this problem to make clear that DHT is imbalanced essentially. We gave the precise scope of the upper bound of imbalance, and verified the results with simulations.

Keywords Peer-to-peer network, Distributed hash table (DHT), Load balance, Probability distribution

对等网(P2P)由于具有支持大规模分布式计算和存储的潜质,成为当前计算机网络领域的一个研究热点。对等网的一个核心问题是如何把计算任务或者存储对象分发到各个节点上,分布式哈希表(DHT)算法作为这个问题的一个非常有竞争力的解决方案,近年来被广泛研究[1-3]。

在DHT中,通常采用某种Hash函数为节点分配ID,这个Hash函数能够保证节点均匀地分布在整个地址空间。对象的ID分配最简单的处理方案是采用同样的Hash函数,但是由于Hash函数会破坏对象之间的关联,所以在类似于数据库这样的应用中,往往会采用其它类型的函数。在不同的DHT算法中,对象映射到节点上的方法也不尽相同[1-3],由于地址空间足够大,节点在地址空间中的分布相对比较稀疏,所以对象通常是映射到距离最近的节点上。

DHT算法的一个重要议题是负载均衡问题,但是这个问题很容易被误解,其可能被认为:节点的ID是由某个Hash函数计算而来,所以就能够保证节点在地址空间里均匀分布。但是需要注意的是,与负载均衡性相关的是节点间的距离如何分布,而不是节点如何分布。一个节点与相邻节点间的距离越长,对象映射到其上的概率就越大。而分析[4]表明,虽然节点在地址空间上均匀分布,但是节点间的距离却呈现几何分布。

本文的目的是系统地对DHT负载均衡的必要性进行总结,澄清对这一问题的误解,以引起人们对该问题足够的重视;同时给出负载均衡上界的范围,并利用仿真实验加以验证。

1 DHT负载均衡

负载是一个与服务相关的概念,如果对服务进行量化,就可以统计出各个节点的负载。量化的度量可以选择访问次数或者计算时间等,本文简单地选取消息的个数作为负载度量。

对于DHT来说,它提供3种类似于Hash表的操作:put,get与delete。其中,put操作把对象存放到DHT中的节点上;对应地,delete操作从节点上删除对象;而get操作则是从节点上获取某个对象。通常的应用中,put和delete操作使用的频率较小;但是对应用于存储系统的DHT而言,由于要周期地发布对象,put操作的使用频率比较高[5]。此外,由于各个节点还承担路由的功能,因此有些学者[6,7]更进一步地考虑了节点的路由负载情况。同时,每个节点本身还有路由表的维护开销。因此,对于DHT算法而言,一个节点接收到的消息主要有4种:get消息、put消息、转发消息与路由维护消息。其中,get消息的个数是存储在其上的所有对象的访问次数的总和。这与两个方面的因素相关:首先是节点上存储对象的个数,其次是对象的热度(popularity)。前者主要取决于对象是否大致相等地分配在所有节点上,后者是由DHT算法的应用场合决定。特别地,对于后者来说,对象的查询热度一般服从Zipf分布[8,9]。put消息个数则与该节点所存储

到稿日期:2008-10-27 返修日期:2009-03-20

聂晓文(1972-),男,博士研究生,主要研究方向为P2P计算,E-mail:niexiaowen@uestc.edu.cn;卢显良(1944-),男,教授,博士生导师,主要研究方向为分布式存储与高级操作系统;李 梁(1982-),男,博士研究生,主要研究方向为P2P计算;徐海湄(1975-),女,博士研究生,主要研究方向为P2P计算;蒲 汛(1976-),男,博士研究生,主要研究方向为网格计算。

的对象个数相关。转发消息直接与路由表的设计相关，而且间接受查询对象的 Zipf 分布影响。路由表的维护在惰性更新策略下，基本上是一个恒定的开销[10]，不随网络规模的增大而增加。

由于对象查询的 Zipf 分布受具体的应用所影响，而且文献[8,9]已经对此有过讨论，因此排除对象查询的 Zipf 分布这一因素。本文假设对象的分布是均匀的，同时对节点的查询热度也是相同的。这样，DHT 中节点的负载主要与其上分布的对象个数相关。

DHT 负载均衡的出发点是维护系统的公平性与可扩展性。如果节点上的负载不均衡，则对于系统中承担较大负载的节点而言，算法就有失公平；DHT 算法一般都是应用于开放的环境中，不公平的算法会减弱对用户的吸引力。更重要的是，负载不均衡可能会影响到整个系统的可扩展性；随着系统规模的增大，如果这种不平衡加剧，甚至有可能导致过热节点最终崩溃。

DHT 负载均衡的最终目标是使所有节点的负载相同——这个目标不太可能现实，所以可行的目标是使所有节点的负载以高概率分布在某个窄小的区间内。

2 理论分析

假设对象 ID 采用某种 Hash 函数进行分配，所有对象在地址空间上均匀分布，节点的负载就与其上存放的对象个数相关；而对象个数又正比于该节点管理的地址空间范围。所以，节点的负载是否均衡与节点管理的地址空间范围是否彼此相等直接相关。

设 n 为系统中节点个数，而 N 表示地址空间大小。节点 ID 的分配由于采用 Hash 函数，保证了节点在整个地址空间均匀分布，但是不能错误地认为节点管理的地址空间都大致等于 $1/n$。Stoica 等[2]最先认识到这一点，并给出了结论：系统中，节点拥有最大地址空间是拥有的平均地址空间的 $\log(n)$ 倍，记为 $f_{\max}$。而文献[11,12]则给出了结论：系统中节点拥有的平均地址空间是拥有的最小地址空间的 n 倍，记为 $f_{\min}$。Wang[13]更进一步证明了节点管理的最大地址空间和最小地址空间的取值范围为：

$$\log n-\log(c\log n)\leqslant f_{\max}\leqslant(1+c)\log n$$

$$\frac{n}{\log n}\leqslant f_{\min}\leqslant n^{1+c}$$

以上的工作说明了节点负载在一个范围内变化，但是如果节点负载在这个范围类似于正态分布，我们仍然有理由不去理睬负载均衡问题。King[14]通过仿真实验验证了节点管理范围的分布类似于某种负指数分布：总的来说，节点的分布偏向于小范围地址空间；换句话说，管理小范围地址空间的节点占大多数。

本文采用 Chord 网络作为讨论模型，关于 Chord 网络的详细介绍请参阅相关文献。在后文的讨论中，做出如下假设：首先，假设节点地址空间为连续的单位圆环，即节点 ID 属于区间[0,1)；其次假设节点 ID 随机选取，即如果把节点 ID 看成一个随机变量，则 ID 服从均匀分布，$id\sim U[0,1)$，$U[0,1)$ 表示在区间[0,1)上的均匀分布。根据 Chord，节点或位置 p 的直接后继节点记为 $suc(p)$，p 的第 $i>0$ 个后继节点记为 $suc^i(p)$。

在以上假设下，讨论节点之间的间距以及在一段地址空间上的节点个数所服从的概率分布。

定理 1 有一条长为 L 的线段，在其上随机放置 $n-1$ 个节点，使之切分为 n 条小线段。在这些小线段中任选一条，设其长度为 l。则随机变量 l 的概率密度为 $f(l)=(n-1)\frac{(L-l)^{n-2}}{L^{n-1}}$。

证明：如图 1 所示，首先 n 条小线段长度的分布是相同的，所以只需考虑最右端或最左端的小线段分布即可。

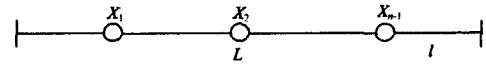

图 1 k 个点分割一条线段示意图

考察最右端小线段长度 l 的概率分布。它的长度与第 $n-1$ 个节点的位置 X 有关，$l=L-X$，而 X 是 $n-1$ 个节点中坐标最大的节点。这些随机放置的节点的坐标在[0,L]上服从均匀分布，即 $X_i\sim U[0,L]$，它们的 CDF 为 $F(x)=x/L$。而图 1 中最右端的小线段的左端点是这些节点中位置最大的节点，所以 $X=\max\{X_1,X_2,\cdots,X_{n-1}\}$。由于 $\max\{\}$ 函数的 CDF 是 X_i 的 CDF 的 $n-1$ 次方，因此最右端小线段的左端点的 CDF 和 PDF 为：

$$F(x)=\left(\frac{x}{L}\right)^{n-1} \tag{1}$$

$$f(x)=(n-1)\frac{x^{n-2}}{L^{n-1}} \tag{2}$$

再根据 $l=L-X$，计算 l 的 CDF 和 PDF：

$$F(l)=1-\left(\frac{L-l}{L}\right)^{n-1} \tag{3}$$

$$f(l)=(n-1)\frac{(L-l)^{n-2}}{L^{n-1}} \tag{4}$$

命题得证。

在单位圆周上随机放置 n 个节点，把单位圆周切分为 n 条圆弧。设某条圆弧的长度为 l，把整个圆周在第一个节点处分割成长度为 1 的线段。根据定理 1，则 l 的 CDF 和 PDF 为：

$$F(l)=1-(1-l)^{n-1} \tag{5}$$

$$f(l)=(n-1)(1-l)^{n-2} \tag{6}$$

引理 1 设两个函数 $f_1(x)=a-[g_1(x)]^k$，$f_2(x)=a-[g_2(x)]^k$，其中 a 和 k 为常数，并且 $|a|>1$，$k>1$；而 x 取值保证 $g_1(x)$ 和 $g_2(x)$ 有界，$0\leqslant g_1(x),g_2(x)<1$；则 $k\to\infty$ 时，$f_1(x)/f_2(x)\to 1$。

证明：设 $h^1_{\max}=\max\{g_1(x)\}$，$h^2_{\max}=\max\{g_2(x)\}$，$h^1_{\min}=\min\{g_1(x)\}$ 和 $h^2_{\min}=\min\{g_2(x)\}$。令 $\Delta(x)=\frac{f_1(x)}{f_2(x)}$，则：$\frac{a-(h^1_{\max})^k}{a-(h^2_{\min})^k}\leqslant\Delta(x)\leqslant\frac{a-(h^1_{\min})^k}{a-(h^2_{\max})^k}$。当 $k\to\infty$ 时，由于 $0\leqslant h<1$，故 $h\to 0$，所以 $\Delta(x)\to 1$。

定理 2 当 n 足够大时，式(5)近似于指数分布 $F(l)=1-e^{-nl}$。

证明：根据引理 1，当 $0<l<1$ 时，命题成立。现讨论 $l\to 0$ 时的情况，此时 $1-e^{-nl}\to 0$，$1-(1-l)^{n-1}\to 0$。而 $\lim_{l\to 0}\frac{1-(1-l)^{n-1}}{1-e^{-nl}}=\frac{n-1}{n}$，所以命题成立。

根据定理 2，当 n 非常大时，完全可以使用指数分布来近

似表示式(5)和式(6)的分布。图 2 是当 $n=10^3,\cdots,10^4$，式(5)与指数分布的 CDF 函数之比的曲面图。从图 2 中可以观察到函数比值都落在区间(0.999,1.0002)上；并且随着 n 的增加，比值越接近于 1，如图 2(b)中所示的 $n=10^4$ 时，比值落在(0.9999,1.0002)上。

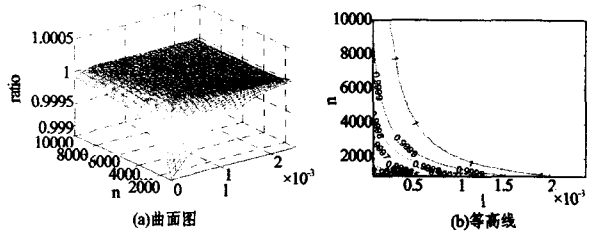

图 2　式(5)与指数分布比值的曲面图

在 DHT 网络中，落在节点上的资源数量正比于节点的管理范围(Zone)大小，亦即节点负载正比于 Zone 的大小。而在 Chord 网络中，节点的管理范围(Zone)定义为一个节点与前驱之间的间距，但不包括前驱节点本身。根据定理 2，在 Chord 网络中 Zone 的分布为一个指数分布，其概率密度(PDF)为 $f(l)=ne^{-nl}$，均值为 $E[l]=\frac{1}{n}$，方差为 $D[l]=\frac{1}{n^2}$。

文献[13]定义 f_{max} 表示网络中最长间距与平均间距长度的比值，定义 f_{min} 表示平均间距与最短间距的比值，本节沿用这种定义，下面求解这两个值。

由于累积分布函数(CDF)$F(l)$ 表示所有小于 l 的 Zone 的概率，而网络规模为 n，所以 $nF(l)$ 就表示所有小于 l 的 Zone 的似然个数。令 $nF(l)\leqslant 1$，则 $nF(l)$ 表示小于 l 的 Zone 的个数为 1，求解即得到网络中最小间距(l_{min})的值：

$$l_{min}\leqslant\frac{\ln(\frac{n}{n-1})}{n} \tag{7}$$

而 Zone 的平均值为 $1/n$，所以：

$$f_{min}\geqslant\frac{1}{\ln(\frac{n}{n-1})} \tag{8}$$

再估算 f_{min} 的双侧置信区间。设置信度为 95%，则 f_{min} 置信区间的右侧对应于 $nF(l)\leqslant 0.1$，置信区间的左侧对应于 $nF(l)\leqslant 10$，分别求解得到置信度为 $1-\alpha=95\%$ 的双侧置信区间大致为：

$$\left[\ln(\frac{n}{n-10})^{-1},\ln(\frac{n}{n-0.1})^{-1}\right] \tag{9}$$

根据定理 2，所有大于 l 的 Zone 的概率为 e^{-nl}，则 ne^{-nl} 表示所有大于 l 的 Zone 的似然个数，令其为 1，求解得到网络中最大间距(l_{max})的值为：

$$l_{max}\geqslant\frac{\ln(n)}{n} \tag{10}$$

$$f_{max}\geqslant\ln(n) \tag{11}$$

再估算 f_{max} 置信度为 95%的双侧置信区间为：

$$[\ln(0.1n),\ln(10n)] \tag{12}$$

对式(8)的右侧求 n 的导数，得到：

$$\frac{1}{n(n-1)\ln(\frac{n}{n-1})^2} \tag{13}$$

这是一个正数；对式(11)右侧求导，得到 $1/n$；所以 f_{min} 和 f_{max} 随着 n 的增加而增加。这说明随着网络规模的增长，Chord 网络中 Zone 负载的不平衡性也在增加。

因此，DHT 网络中通常采用哈希函数给节点分配 ID，这使得节点在整个地址空间上均匀分布，但是这并不意味着节点的 Zone 负载是均衡的。理论分析表明，Chord 网络中节点 Zone 负载是不均衡的，这暗示着其它 DHT 算法都应该检查 Zone 负载的均衡情况。

3　仿真实验

为验证以上结论，选取地址空间为 2^{32}，网络规模为 $n=10^5$ 进行仿真实验。实验开始时，向网络中加入 $n=10^5$ 个节点，然后不停地加入新节点，并从在线节点中随机选取节点退出，保持网络规模不变。在系统稳定后，提取节点间距进行分析。需要注意的是，退出节点的选择不能通过先生成一个随机值，然后找到该值的直接后继作为退出节点；这样的方法会使在线节点的退出概率不等[14]。

图 3 是仿真实验的间距分布图，横坐标表示间距长度，纵坐标是表示对应的间距个数。图 3 中的理论值与实测值非常吻合，说明了式(6)的结论是正确的。

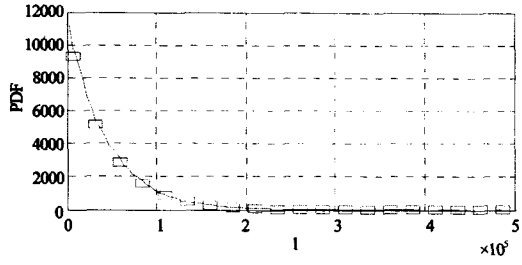

图 3　仿真实验结果

在图 3 的仿真测试中，利用式(10)来推算 l_{max} 的上界，这个值为 $4.9448*10^5$，利用式(12)计算得到 l_{max} 的 95%置信区间为$[3.9558*10^5,5.9337*10^5]$。实际测试中，最大的 l 为 505840，排在第 10 的 l 为 391413。对比理论的计算值，l_{max} 的估计误差是相当小的。

根据式(8)，计算 l_{min} 为 0.4295，再根据式(9)计算 l_{min} 的 95%置信区间为[0.0429,4.2952]。由于测试的地址空间为离散，最小的 l 为 1。图 3 的实际测试中，$l=1$ 的情况出现了 4 次，$l=3$ 的情况出现了 1 次，$l=5$ 的情况出现了 5 次。对比理论值，可以发现理论计算值也还是比较准确的。

从图 3 可以直观地了解到，节点间距的分布呈现负指数类型的分布；节点间距越小，在网络中出现的概率越大。图 3 直观地反映了 Chord 网络中负载本质上是负载不均衡的。

结束语　DHT 的 ID 值分配往往采用 Hash 函数，这能够保证节点在整个地址空间中均匀分布，但是这并不意味节点间距向均值集中。实际的情况是，节点间距呈指数分布，节点间距越小，概率越大。因此，DHT 的负载本质上是不均衡的，而现有的 DHT 应用往往对这一点认识不足或者不够重视。通过对节点间距的分析，可以清楚了解到 DHT 算法在应用中必须要配合相应的负载均衡算法，否则难以保证 DHT 算法的公平性和效率。在通常的 DHT 算法中附加负载均衡算法是必要的，而且是必需的。

参 考 文 献

[1] Zhao B Y, Kubiatowicz J D, Joseph A D. Tapestry: An Infrastructure for Fault-tolerant Wide-area Location[D]. University of California at Berkeley, 2001

[2] Stoica I, et al. Chord: a scalable peer-to-peer lookup protocol for Internet applications [J]. Networking, IEEE/ACM Transactions, 2003, 11(1): 17-32

[3] Rowstron A I T, Druschel P. Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems [C]// Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms Heidelberg. Springer-Verlag, 2001

[4] Krishnamurthy S, et al. A Statistical Theory of Chord under Churn[C]// Proceedings of The 4th Annual International Workshop on Peer-To-Peer Systems (IPTPS 05). Ithaca, NY, USA, 2005

[5] Mo Z, Yafei D, Xiaoming L. A measurement study of the structured overlay network in P2P file-sharing systems[M]. Hindawi Publishing Corp, 2007

[6] Silvia Bianchi S S, Felber P, Kropf P. Adaptive Load Balancing for DHT Lookups[C]// Proceedings of the 15th International Conference on Computer Communications and Networks (ICCCN'06). Arlington, VA, October 2006

[7] Datta A, Schmidt R, Aberer K. Query-load balancing in structured overlays[C]// Proceedings of the Seventh IEEE International Symposium on Cluster Computing and the Grid (CCGRID'07). Riode Janeiro, Brazil, 2007

[8] Baeza-Yates R, Ribeiro-Neto B. Modern Information Retrieval [M]. Addison Wesley, 1999

[9] Saroiu S G, Krishna, Steven G. A Measurement Study of Peer-to-Peer File Sharing Systems[C]// Proceedings of Multimedia Computing and Networking 2002 (MMCN '02). 2002

[10] Rhea S, et al. Handling Churn in a DHT[D]. EECS Department, University of California, Berkeley, 2003

[11] Moni N, Udi W. Novel architectures for P2P applications: the continuous-discrete approach[C]// Proceedings of the fifteenth annual ACM symposium on Parallel algorithms and architectures. ACM: San Diego, California, USA, 2003

[12] Valerie K, Jared S. Choosing a random peer[C]// Proceedings of the twenty-third annual ACM symposium on Principles of distributed computing. ACM: St. John's, Newfoundland, Canada, 2004

[13] Xiaoming W, Dmitri L. Load-balancing performance of consistent hashing: asymptotic analysis of random node join [M]. IEEE Press, 2007

[14] Valerie K, et al. Choosing a Random Peer in Chord[M]. Springer-Verlag New York, 2007

(上接第 50 页)

从表 2 的测试数据中可以看出，SCCF 算法在 4 种复杂网络中均能返回具有最高 Q 值的社团结构，这说明用 SCCF 算法划分得到的社团结构是最令人满意的。而 GN，NM 和 MS 算法由于超额的系统资源占用，在由 BBS 数据构成的大型虚拟社会网络中甚至没能完成实验。

从表 3 可以看出，GN，NM 和 WS 算法在仅含有几百个节点的小型网络中能够有效地进行社团结构探测，但在节点数上万的虚拟社会网络中探测时，这 3 种算法均失效。而 SCCF 算法却可以在很短的时间内处理具有上万个节点的大型网络。

结束语 复杂网络的社团发现已经成为当今一个非常具有挑战性和前景性的研究领域。本文用基于谱聚类算法的思想来优化网络社团结构的划分。与已有的社团发现算法相比，本算法时间复杂度较低，且能够在结构未知的大型网络中得到高质量的社团结构。

参 考 文 献

[1] Strogatz S H. Exploring complex networks [J]. Nature, 2001, 410: 268-276

[2] Garey M R, Johnson D S. Computers and Intractability: A Guide to the Theory of NP-Completeness[M]. San Francisco: W. H Freeman Publishers, 1979

[3] Scott J. Social Network Analysis: A Handbook [M]. 2 nd ed. London: Sage Publications, 2002

[4] Fiedler M. Algebraic connectivity of graphs[J]. Czech, Math J, 1973, 23: 298-305

[5] Shi J, Malik J. Normalized cuts and image segmentation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000, 22(8): 888-905

[6] Gu M, Zha H, Ding C, et al. Spectral relaxation models and structure analysis for k-way graph clustering and bi-clustering [R]. CSE-01-007. Penn State University, 2001

[7] Meila M, Shi J. Learning segmentation by random walks[C]// NIPS. 2000: 873-879

[8] Muff S, Rao F, Cflisch A. Validation of network clustrizations [J]. arXiv: cond-mat, 2005: 0503252

[9] Newman M E J, Girvan M. Finding and evaluating community structure in networks[J]. Physical Review E., 2004, 69(2): 026113

[10] White S, Smyth P. A spectral clustering approach to finding communities in graph[J]. SIAM Data Mining, 2005

[11] Zachary W. An information flow model of conflict and fission in small groups[J]. J. Anthropol. Res., 1993, 33: 452-473

[12] Girvan M, Newman M. Community structure in social and biological networks[J]. Proc Natl Acad Sci USA, 2002, 99: 7821-7826

[13] Gleiser P, Danon L. Community structure in jazz. Advances in Complex Systems, 2003, 6: 565

[14] Newman M. Modularity and community structure in networks [J]. Proc Natl Acad Sci USA, 2006, 103: 8577-8582

[15] Donetti L, Munoz M A. Detecting Network Communities: a new systematic and efficient algorithm[J]. cond -mat, 2004: 0404652