

动态结构化 P2P 网络的负载均衡方案*

彭利民¹ 肖文俊²

(1. 华南理工大学 计算机科学与工程学院, 广东 广州 510006; 2. 华南理工大学 软件学院, 广东 广州 510006)

摘 要: DHT 结构化 P2P 网络中, 节点上存放对象个数的差异性、节点处理能力的异构性以及 P2P 网络的动态性, 使 P2P 网络出现负载不均衡问题. 为此, 文中在超立方 DHT 覆盖网络上构建了基于二叉树的层次化负载均衡模型, 用于收集节点的负载和容量信息、生成负载均衡策略和执行负载转移操作. 通过应用均衡域的操作模式, P2P 网络的负载均衡任务可按照并行与分布式进行处理. 模拟结果表明, 在动态的 P2P 网络环境下, 负载均衡方案可获得较好的效果.

关键词: P2P 网络; 负载均衡; 虚拟服务器; 均衡域; 二叉树

中图分类号: TP393

doi: 10.3969/j.issn.1000-565X.2011.10.010

基于分布式散列表(DHT)的 P2P 网络, 通过采用统一的 Hash 函数将网络中的对象和节点映射到一个键值空间, 然后将资源按照键值存储在键值相等或相近的节点上. 由于 Hash 函数的随机性, 每个键值空间中的节点和对象分布也存在随机性, 因此, 每个节点负责的对象个数也可能不同, 文献[1]中已证明: 结构化 P2P 网络中某些节点负责的标识符空间可能是其它节点的 $O(\log_2 N)$ 倍, N 为 P2P 网络的节点数. 另外, P2P 网络中所有节点在分配存储对象或负载时, 不论其处理能力是否相同, 都承担着同样的功能角色. 现有研究表明: P2P 网络中节点的处理能力(包括存储空间、带宽及 CPU 性能等)具有很大的差异性^[2], 容易出现某些处理能力较弱的节点承担较高的负载, 而处理能力较强的节点承担较低的负载, 使 P2P 网络呈现负载不均衡问题.

负载均衡是一个经典且被广泛研究的课题, 文献[3]中针对并行处理系统提出的负载均衡算法, 有效地将单个处理任务有机地分配到多个处理器上, 减少多处理器系统中单任务的执行时间. 为实现分布式系统的动态负载平衡, 文献[4]中基于 Multi-Agent 提出了一种新的分布式系统动态负载平衡算

法. 根据节点间转移虚拟服务器的思想, 文献[5]中针对 P2P 网络的负载不均衡问题, 提出了一对一、一对多和多对多负载均衡算法. 文献[6]中扩展了文献[5]的一对多和多对多负载均衡算法, 使其适应动态 P2P 系统. 但文献[5-6]中的负载均衡算法依赖系统中指定的 d 个目录服务节点来收集负载信息和生成转移策略, 这种类似于集中式处理方式容易引起单点失效问题; 而且在转移负载时, 由于没有考虑节点间的物理位置关系, 使负载均衡开销增大, 延缓了负载均衡操作的收敛时间. 文献[7]中通过在 Chord 系统上嵌入 K-ary 树模型, 并采用界标簇算法来收集网络中节点的位置信息, 使负载转移尽量在物理位置较近的节点之间进行, 从而减少转移负载的物理跳数, 节省系统资源. 但 K-ary 树的根节点需要定期收集所有节点的负载信息, 并在将收到的信息分发给 K-ary 树中各个节点后, 模型中的节点才能确定自身的负载状态. 因此, 在 K-ary 树中某个父节点失效到其恢复前, 其孩子节点所在子树中的负载不均衡问题无法解决, 并且每次负载转移后, K-ary 树需要重新构造, 使得构造和维护 K-ary 树的开销较大, P2P 系统不便于扩展. 在文献[8]中, 每

收稿日期: 2010-12-21

* 基金项目: 国家自然科学基金资助项目(60973150, 61103037); 国家科技支撑计划项目(2009BAH46B01)

作者简介: 彭利民(1976-), 男, 在职博士生, 广州体育学院讲师, 主要从事分布式计算研究. E-mail: penglm86@126.com

个节点周期性地收集邻近区域内其它节点的负载信息,并选择链路延迟较小的节点转移负载,但该方法只能保证局部的负载均衡,很难较快地使整个 P2P 系统达到负载均衡。

文中在超立方体 P2P 覆盖网络上构建一个基于二叉树的负载均衡模型,根据节点的承载容量分配相应的负载,以减少负载均衡过程中的通信冗余和负载转移开销等。首先,将 P2P 系统中的节点组织成一个层次化的二叉树型结构,负责收集与分发节点的负载信息,以减少负载均衡代价,并便于 P2P 系统的扩展。同时,通过引入均衡域概念,将 P2P 系统中的节点划分到各个均衡域中,使负载均衡操作可以在整个系统或各均衡域中完成,有利于将整个 P2P 系统的负载均衡任务按照并行与分布式处理,降低负载均衡算法的时间复杂度,使负载均衡模型具有很好的适应性。

1 系统模型

1.1 相关概念和定义

(1)虚拟服务器。虚拟服务器是 Chord 系统中为改善节点负载量而提出的一个概念^[9],也是文中负载均衡处理的基本单位。虚拟服务器类似于 P2P 网络中的节点,一个虚拟服务器负责相应的键值空间,而每个物理节点可拥有多个虚拟服务器。从负载均衡的观点,虚拟服务器可以表示确定的负载量。当物理节点过载时,可以在其拥有的虚拟服务器中选择一个或多个虚拟服务器转移到其它非过载节点上。

(2)均衡域。均衡域是并行计算机系统中的一个概念^[3],通过将系统划分为多个独立的处理器集合(称为均衡域),可以将单个处理任务分配到多个处理器上,减少多处理器系统中单任务的执行时间。在图 2 中,均衡域是指位于同一棵子树中的节点集合,如节点 000 和 001 属于同一个均衡域,节点 000、001、010 和 011 属于同一个均衡域。均衡域的规模可以从几个节点到整个系统,负载均衡决策唯一依赖于每个均衡域的负载状态,各个均衡域可以并行地进行负载均衡操作,以减少整个系统负载均衡操作的收敛时间,降低负载均衡算法的时间复杂度。

(3)节点利用率。节点利用率是指节点的负载与承载能力的比值。节点的承载能力(综合处理能力)可以是节点的 CPU 处理能力,也可以是节点的存储空间大小。不失一般性,文中假定其为节点的存储空间大小。通过引入节点利用率的概念,在负载均

衡过程中可有效地处理 P2P 系统中的节点异构性问题,并按照节点的实际承载能力分配相应的负载,从而避免承载能力弱的节点承担较高的负载量,使 P2P 系统中的负载均衡分布。

(4)均衡域利用率。均衡域利用率是指均衡域中所有节点的负载总量与总承载能力的比值。当整个系统属于同一个均衡域中时,均衡域利用率表示系统利用率。根据均衡域利用率,可以在不同的均衡域内实施不同的负载均衡策略。

(5)海明距离。海明距离是计算机网络通信理论中的一个概念,两个码字中对应位的比特值不同的位数,即为两个码字的海明距离。由于文中采用二进制数对超立方体 P2P 覆盖网络中的节点进行标识,因此,海明距离也可以用于表示节点间的最小距离。如节点 A(10001001)和 B(10110001)的第 3、4 和 5 位对应的比特值不同,其余的比特值均相同,因此,节点 A 和 B 之间的距离为 3。一般将过载节点上的负载转移到距离较近的节点上,以减少负载均衡的开销。

1.2 基于二叉树的层次负载均衡模型

超立方体结构具有较优的路由算法,其操作与维护简单,因而是 P2P 网络中一个较常用的覆盖网络结构,Pastry^[10]、Tapestry^[11]等都是基于超立方体结构的 P2P 系统。文献[12]中提出的超立方体 DHT 覆盖网络,保持了覆盖拓扑和物理拓扑的一致性,有效地支持 P2P 系统的文件共享。文中在文献[12]的超立方体 DHT 覆盖结构(见图 1)上,建立一个基于二叉树的层次负载均衡模型(见图 2),负责收集与分发 P2P 系统中的负载信息,以及执行负载转移操作,以解决动态 DHT 网络中的负载均衡问题。

图 1 中的 h_0 、 h_1 和 h_2 分别表示超立方体的第 0、1 和 2 维度上的边。二叉树中的中间节点表示其下层

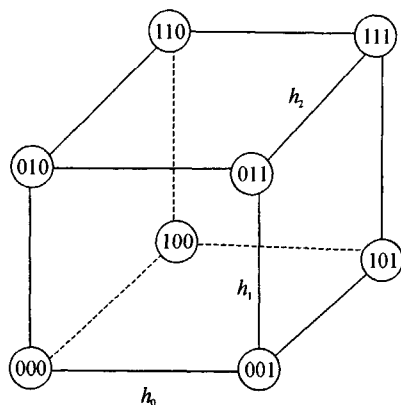


图 1 三维超立方体覆盖网络

Fig. 1 A 3-D hypercube overlay network

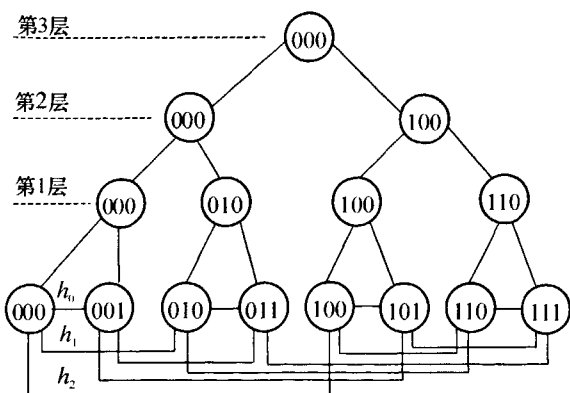


图2 基于二叉树的负载均衡模型

Fig. 2 A binary-tree based load balancing model

相应均衡域中的域头节点,这些节点负责收集自身均衡域内的负载信息以及控制均衡处理过程.例如,第1层的000、010、100和110节点分别负责第0层的2个均衡域(或节点)的负载均衡操作,第2层的000和100节点分别负责其管辖的第1层的2个均衡域的负载均衡操作,第3层的000节点负责其管辖的第2层的2个均衡域的均衡操作过程,按照这种自下而上的层次处理方法,该二叉树模型将系统组织成一个层次的负载均衡域结构,分解了P2P系统的负载均衡处理过程.

根据二叉树负载均衡模型,可以按照自下而上的层次方法,在不同规模的均衡域内中实现负载转移.例如,当节点000超载时,由节点000触发负载均衡事件,由于节点000是由节点000和节点001组成的均衡域中的域头节点,因此,节点000分析节点001是否可以用来转移负载,如果可以,则将节点000上过载的负载转移到节点001上,否则,按照自下而上的层次方法将过载负载转移到第2层或第3层的均衡域内的节点上.另外,由于负载均衡操作可在不同层次的均衡域内进行,而且超立方体结构路由算法简单,所以负载转移易于实现,从而简化负载均衡的操作过程,降低负载均衡的开销和负载转移代价.

2 分布式负载均衡算法

首先每个节点向其父节点发送负载信息,然后域头节点计算每棵子树(均衡域)的负载状态并依次传到树根节点.当某个节点(或均衡域)的节点利用率超过预设的阈值,则触发负载均衡事件.由于DHT网络中节点的异构性,在设计负载均衡算法时,需要考虑节点承载能力的差异性,因此,文中根据节点利用率来计算需要转移的超载量及确定可以接收超载量的节点(或节点集),从而使分布式负载均衡算法便于处理异构DHT网络中的负载均衡问题.

Procedure load balancing algorithm ()

// 在超立方体 DHT 覆盖结构上构造二叉树模型

for $i = 0$ to $n - 1$ do // $n = \log_2 N$

constructDomain() at the level i ;

chooseDomainHead();

end for

// 收集节点的负载信息,计算均衡域的利用率

for $i = 0$ to $n - 1$ do

send(c_i , load $_i$) to domainHead;

computeDomainUtilization();

end for

// 若节点利用率高于预设的阈值,则将其容量信息和虚拟服务器上的负载信息发送给同层的域头节点

if ($u_k > \text{sys_util}$) then

send(c_k , {vsload $_1$, ..., vsload $_m$ }) to domainHead;

// 首先计算节点 k 上的超载负荷量,然后将节点 k 的负载转移到与节点 k 距离最近的节点 v 上

vs $_{\text{over}} = (u_k - \text{sys_util})c_k$;

for $m = 1$ to $n - 1$ do

if $\left(\sum_{j=1}^{2^{m-1}} (\text{sys_util} - u_j)c_j \right) \geq \text{vs}_{\text{over}}$ then

chooseNodes() for transferring load of node k ;

PerformTransfer(k, v) for each transfer $k \rightarrow v$;

end if

end for

end if

对于一个包含 N 个节点的 P2P 系统,负载均衡模型中第 i 层共有 $N/2^i$ 个均衡域.当所有均衡域内的负载都不均衡时(最坏的情形下),每个均衡域内最多发送的负载均衡请求信息和负载转移次数均为 2^{i-1} ,则每层中负载均衡操作次数最多为 $N/2$.由于二叉树模型共有 $\log_2 N$ 层,因此,整个 P2P 系统中负载均衡操作次数最多为 $N \log_2 N$.由于 P2P 系统的均衡域可以按照平均并行度为 $N/2$ 的并行方式执行负载均衡操作,因此,算法的时间复杂度为 $O(\log_2 N)$.

3 仿真实验与结果分析

实验使用 P2PSim 作为模拟器来评估文中提出的负载均衡算法.为了便于比较,文中同时实现了文献[6]中提出的负载均衡算法,表1列出了实验参数值.实验采用 P2P 系统中3个主要的标度进行测定,即节点利用率、负载转移因子和负载转移跳步数.其中,负载转移因子是指负载均衡过程中的负载转移代价与系统中所有负载转移一次的总代价的比值^[6],它表示负载均衡过程中的负载转移代价.负载转移跳步数是指在负载均衡过程中,负载从超载节点转移到轻载节点路由过程中的物理跳步数,它表

示负载均衡开销,负载转移跳步数越小,负载均衡开销也越小. 每次模拟时间均为 $20T$, 其中, T 表示预设的均衡周期. 对于每个实验情况进行 10 次实验, 取 10 次实验结果的平均值作为该实验的最终结果.

表 1 实验参数设置

Table 1 Setting of experimental parameters

环境参数	默认值
系统利用率/%	85
对象分布	均匀地分布在标识符空间
对象个数	10^6
对象负载分布	Pareto($\alpha=2, \theta=0.5$)
节点个数	4096
节点的承载容量分布	Pareto($\alpha=2, \theta=100$)
负载均衡周期/s	60
每个节点上的虚拟服务器个数	12
目录节点个数	16

实验 1 测试在不同的系统利用率下, 执行负载均衡后 P2P 系统中的节点利用率分布情况, 结果如图 3 所示.

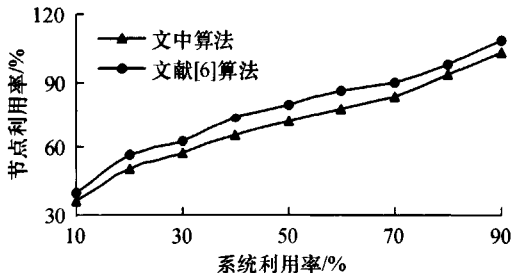


图 3 负载均衡后节点利用率分布

Fig. 3 Distribution of node utilization rate after load balancing

图 3 表明: 负载均衡后, 两种负载均衡算法的节点利用率随系统利用率增加呈近似线性递增趋势, 且负载均衡后的节点利用率略大于系统利用率. 由于文献[6]中的算法由多个目录节点负责整个系统的负载均衡操作, 各个目录节点之间相互独立, 一个目录节点管辖内节点上的负载无法转移到另一个目录节点管辖的节点中, 因此, 无法使整个 P2P 系统达到负载均衡. 在文中提出的负载均衡方案中, 各个节点按照均衡域的方式进行组织, 且均衡域可以按照由下至上的层次结构进行扩展至整个系统, 因此, 可以使整个 P2P 系统达到一致性的负载均衡. 从图 3 可以看出, 文中负载均衡算法的节点利用率均低于文献[6]中算法的节点利用率, 说明文中算法比文献[6]中的算法能取得更好的负载均衡效果, 当系统利用率为 90% 时, 文中算法使网络中 93% 的节点利用率低于 100%, 89% 的节点利用率低于 90%.

实验 2 测试在不同的系统利用率下, 负载均衡过程中的负载转移因子分布情况, 结果见图 4.

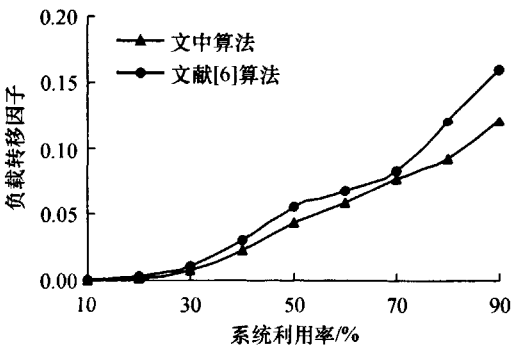


图 4 负载均衡过程中的负载转移因子

Fig. 4 Load movement factor during load balancing

图 4 表明: 在不同的系统利用率下, 两种算法的负载转移因子均随系统利用率增加而增大, 且两种算法的负载转移因子均较小, 说明两种算法在负载均衡过程中, 只需要转移较小的负载量就能使 P2P 系统中的负载均衡分布. 由于文中的负载均衡算法以均衡域模式进行操作, 负载均衡可以在局部范围内执行, 然后再扩展至整个系统; 当某个节点超载时, 文献[6]中的算法首先随机地选择 d 个目录节点, 再由目录节点在其管辖的节点范围内执行负载均衡操作. 因此, 对于不同的系统利用率, 文中的负载均衡算法总保持较好的性能, 在系统利用率为 90% 时, 文中算法的负载均衡因子小于 0.13, 而文献[6]中算法的负载转移因子约为 0.17.

实验 3 测试负载均衡算法在 P2P 网络动态环境下的负载均衡效果. 主要考察额外的 10% 系统负载总量快速到达系统对负载均衡算法的影响, 并将这额外的 10% 负载量随机地分布在 P2P 系统的标识符空间上, 额外增加的 10% 负载量不仅使节点的负载在短期内的分布更加不均衡, 而且使 P2P 系统中节点承担更大的负载量, 结果见图 5.

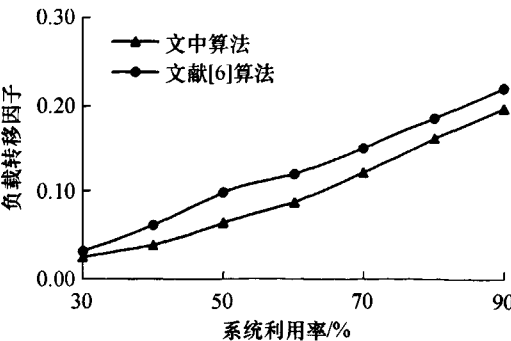


图 5 数据项动态环境下的负载转移因子

Fig. 5 Load movement factor under dynamism of data items

图5表明:负载转移因子随系统利用率增加而呈线性递增.当系统利用率较低时,额外增加的10%负载只造成系统中少量的节点超载,两种算法的负载转移因子均较低.当系统利用率较高时,系统中大部分节点的可承载容量较小,额外增加的10%负载使系统中超载的节点数增加,因此,负载转移因子增大.由于文中算法以均衡域方式执行负载均衡操作,各个均衡域规模可以根据负载状态动态地进行扩展,因此,在各种系统利用率的情况下,文中算法的负载转移因子均比文献[6]中算法小.在系统利用率为90%时,文中算法的负载转移因子小于0.20,而文献[6]中算法的为0.22,表明文中算法能有效地解决动态P2P网络环境下的负载均衡问题.

实验4 测试负载均衡算法在节点动态进入/退出P2P系统情况下的负载均衡效果.P2P系统中节点总数固定为4096,每间隔1s有一个节点随机地进入/退出P2P系统.为了评估负载均衡算法的动态适应性,文中将负载转移因子定义为负载均衡算法所引起的负载转移量与节点到达或离开而产生的负载转移量的比值,结果如图6所示.

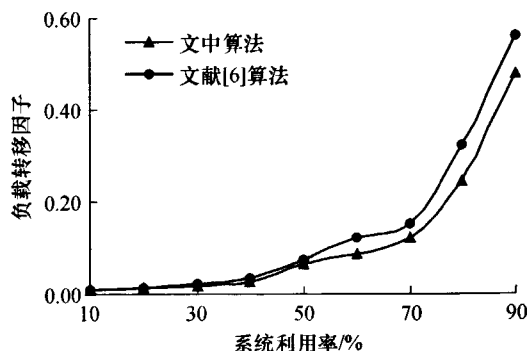


图6 节点动态性与负载转移因子

Fig.6 Load movement factor under dynamism of node

图6表明:负载转移因子随系统利用率增加而增大.当系统利用率较低时,两种算法的负载转移因子均较低,当系统利用率较大时,系统中节点的可承载容量较小,节点随机地加入或退出使P2P系统中的节点负载状态变化增大,需要转移的负载量增多,因此,负载转移因子增大.在系统利用率为90%时,文中算法的负载转移因子小于0.50,文献[8]中算法的负载转移因子为0.56,表明文中算法能有效地解决动态P2P网络环境下的负载均衡问题.

实验5 测试负载均衡过程中转移的负载在物理网络上的跳步距离,以评估负载均衡算法的负载均衡开销.

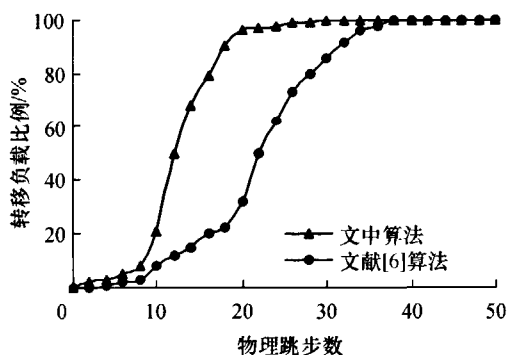


图7 转移负载的物理跳步数累积分布

Fig.7 Cumulative distribution of moved load by physical distance hops

图7显示了两种算法在负载均衡过程中负载转移跳步数的分布情况.在文中的负载均衡模型中,各个均衡域按照节点的位置关系进行组织,而且在各个均衡域中分别地执行负载均衡操作,因此,超载节点和轻载节点都位于同一个均衡域内.当同一个均衡域中有多个轻载节点可用于转移负载时,文中算法总选择距离超载节点最近的节点进行负载转移.在文献[6]算法中,当P2P系统中某个节点 k 超载时,节点 k 在P2P系统中随机地选择目录节点 d ,然后由目录节点 d 负责执行负载均衡操作,由于目录节点 d 在其管辖范围内的轻载节点中随机地选择轻载节点 h 用于转移负载,因此,过载节点 k 与轻载节点 h 相距离可能较远.从图7可以看出,文中算法使50%的转移负载量在12跳内实现转移,90%的转移负载量在18跳内实现转移,文献[6]中算法在12跳内转移13%的负载,18跳内转移22%的负载量,50%的负载量在22跳内的节点间实现转移.图7的结果表明:文中算法能在邻近节点间实现负载转移,大大地降低了负载均衡的开销.

4 结语

针对结构化动态P2P系统的负载不均衡问题,文中通过建立二叉树负载均衡模型,提出了一种分布式负载均衡算法.仿真结果表明:文中提出的负载均衡算法不仅能解决不同系统负载状态下的负载均衡问题,而且能有效地应对动态环境下负载的快速变化,有效地解决动态DHT网络环境下的负载均衡问题.文中算法在负载转移过程中,考虑了参与转移负载的节点之间的位置关系,使负载在邻近的节点间实现转移,有效地降低了负载均衡的开销.

参考文献:

- [1] Karger D, Lehman E, Leighton T, et al. Consistent hashing and random trees: distributed caching protocols for relie-

- ving hot spots on the World Wide Web [C] // Proceedings of the 29th Annual ACM Symposium on Theory of Computing. Texas: ACM, 1997: 654-663.
- [2] Saroiu S, Gummadi P K, Gribble S D. A measurement study of peer-to-peer file sharing systems [C] // Proceedings of Multimedia Computing and Networking. San Jose: SPIE, 2002: 156-170.
- [3] Willebeek L H, Reeves A P. Strategies for dynamic load balancing on highly parallel computers [J]. IEEE Transactions on Parallel and Distributed Systems, 1993, 9(4): 979-993.
- [4] 闫钧华, 张焕春, 经亚枝. 基于 Multi-agent 的分布式系统负载均衡 [J]. 华南理工大学学报: 自然科学版, 2004, 32(12): 74-79.
Yan Jun-hu, Zhang Huan-chun, Jing Ya-zhi. Load balancing of the distributed system based on multi-agent [J]. Journal of South China University of Technology: Natural Science Edition, 2004, 32(12): 74-79.
- [5] Rao A, Lakshminarayanan K, Surana S, et al. Load balancing in structured P2P systems [C] // Proceedings of the 2nd International Workshop Peer-to-Peer Systems. Berkeley: Springer-Verlag, 2003: 68-79.
- [6] Godfrey B, Lakshminarayanan K, Surana S, et al. Load balancing in dynamic structured P2P systems [C] // Proceeding of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies. Los Alamitos: IEEE, 2004: 2253-2262.
- [7] Zhu Yingwu, Hu Yiming. Efficient, proximity-aware load balancing for DHT-based P2P systems [J]. IEEE Transactions on Parallel and Distributed Systems, 2005, 16(4): 349-361.
- [8] 李振宇, 谢高岗. 基于 DHT 的 P2P 系统的负载均衡算法 [J]. 计算机研究与发展, 2006, 43(9): 1579-1585.
Li Zhen-yu, Xie Gao-gang. A load balancing algorithm for DHT-based P2P systems [J]. Journal of Computer Research and Development, 2006, 43(9): 1579-1585.
- [9] Stoica I, Morris R, Karger D R, et al. Chord: a scalable peer-to-peer lookup protocol for internet applications [J]. IEEE/ACM Transactions on Networking, 2003, 11(1): 17-32.
- [10] Rowstron A, Druschel P. Pastry: scalable, decentralized object location and routing for large-scale peer-to-peer systems [C] // Proceedings of the 18th IFIP/ACM International Conference on Distributed Systems Platforms. Heidelberg: Springer-Verlag, 2001: 329-350.
- [11] Zhao B Y, Huang L, Stribling J, et al. Tapestry: a resilient global-scale overlay for service deployment [J]. IEEE Journal on Selected Areas in Communications, 2004, 22(1): 41-53.
- [12] Gharib M, Barzegar Z, Habibi J. A novel method for supporting locality in peer-to-peer overlays using hypercube topology [C] // Proceeding of the 2010 International Conference on Intelligent Systems, Modelling and Simulation. Liverpool: IEEE, 2010: 391-395.

Load Balancing Scheme for Dynamic Structured P2P Networks

Peng Li-min¹ Xiao Wen-jun²

(1. School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, Guangdong, China;

2. School of Software Engineering, South China University of Technology, Guangzhou 510006, Guangdong, China)

Abstract: In DHT-based structured P2P (Peer to Peer) networks, load imbalance occurs owing to the non-uniform distribution of objects stored at peer nodes, the heterogeneity nature of node capabilities and the dynamic characteristics of P2P networks. In order to solve this problem, a hierarchical load balancing model based on binary trees is established on the hypercube DHT overlay networks. The proposed model is used to collect the load and capacity information of nodes, develop the load balancing strategy and transfer the overloaded loads. In this model, by using the balancing domain, load balancing can be implemented in a parallel and distributed way. Simulation results show that the proposed scheme can achieve preferable load balancing in the environment of dynamic P2P networks.

Key words: peer to peer networks; load balancing; virtual server; balancing domain; binary trees

责任编辑: 许花桃

作者：[彭利民](#)，[肖文俊](#)，[Peng Li-min](#)，[Xiao Wen-jun](#)
作者单位：[彭利民, Peng Li-min\(华南理工大学计算机科学与工程学院, 广东广州, 510006\)](#)，[肖文俊, Xiao Wen-jun\(华南理工大学软件学院, 广东广州, 510006\)](#)
刊名：[华南理工大学学报（自然科学版）](#) 
英文刊名：[Journal of South China University of Technology \(Natural Science Edition\)](#)
年，卷(期)：2011, 39(10)
被引用次数：2次

参考文献(12条)

1. [Karger D;Lehman E;Leighton T Consistent hashing and random trees:distributed caching protocols for relieving hot spots on the World Wide Web](#) 1997
2. [Saroiu S;Gummadi P K;Gribble S D A measurement study of peer-to-peer file sharing systems](#) 2002
3. [Willebeek L H;Reeves A P Strategies for dynamic load balancing on highly parallel computers](#) 1993(04)
4. [闫钧华;张焕春;经亚枝 基于Multi-agent的分布式系统负载均衡\[期刊论文\]-华南理工大学学报\(自然科学版\)](#) 2004(12)
5. [Rao A;Lakshminarayanan K;Surana S Load balancing in structured P2P systems\[外文期刊\]](#) 2003
6. [Godfrey B;Lakshminarayanan K;Surana S Load balancing in dynamic structured P2P systems](#) 2004
7. [Zhu Yingwu;Hu Yiming Efficient,proximity-aware load balancing for DHT-based P2P systems\[外文期刊\]](#) 2005(04)
8. [李振宇;谢高岗 基于DHT的P2P系统的负载均衡算法\[期刊论文\]-计算机研究与发展](#) 2006(09)
9. [Stoica I;Morris R;Karger D R Chord:a scalable peer-to-peer lookup protocol for internet applications\[外文期刊\]](#) 2003(01)
10. [Rowstron A;Druschel P Pastry:scalable,decentralized object location and routing for large-scale peer-to-peer systems](#) 2001
11. [Zhao B Y;Huang L;Stribling J Tapestry:a resilient global-scale overlay for service deployment\[外文期刊\]](#) 2004(01)
12. [Gharib M;Barzegar Z;Habibi J A novel method for supporting locality in peer-to-peer overlays using hypercube topology](#) 2010

引证文献(2条)

1. [马剑锋 基于负载均衡的网络考试系统设计\[期刊论文\]-电脑编程技巧与维护](#) 2013(12)
2. [高迎 Balance-Peer:负载均衡的结构化P2P模型\[期刊论文\]-计算机工程与设计](#) 2013(4)

引用本文格式：[彭利民, 肖文俊. Peng Li-min, Xiao Wen-jun 动态结构化P2P网络的负载均衡方案\[期刊论文\]-华南理工大学学报（自然科学版）](#) 2011(10)