

一种基于移动代理的 P2P 系统主动负载均衡算法

李 慧^{1,2}

(1 金陵科技学院 信息技术学院, 江苏 南京 210001; 2 江苏省信息分析工程实验室, 江苏 南京 210001)

摘 要: 负载均衡是 P2P 系统的研究热点, 针对现有负载均衡算法的不足, 提出了一种基于移动代理的主动负载均衡算法. 该算法使用移动代理收集负载信息、生成负载迁移决策, 降低了系统的通信开销, 而且避免了单点失效问题和“颠簸”现象. 通过引入对用户请求的评估和迁移, 预见性的处理负载问题, 提高了系统的服务质量.

关键词: P2P 系统; 移动代理; 负载均衡; 用户请求

中图分类号: TP393

文献标识码: A

文章编号: 1000-7180(2012)11-0092-03

An Active Load Balancing Algorithm for P2P System Based on Mobile Agent

LI Hui^{1,2}

(1 Department of Information Technology, Jinling Institute of Technology, Nanjing 210001, China;

2 Jiangsu Information Analysis Engineering Laboratory, Nanjing 210001, China)

Abstract: Load balancing is the research focus of P2P system. Aiming at the shortage of existed load balancing algorithm, an active load balancing agent based on mobile is proposed. The algorithm collects load information using mobile agent, generates load migration decisions, reducing the system's communication costs, and avoid single point failures and the “bumps” phenomenon. In order to processing load problem predictability, the assessment and migration of user requests are introduced, which improve system quality of service.

Key words: P2P; mobile agent; load balancing; user request

1 引言

随着互联网技术的快速发展, P2P 系统成为发展最快、最广泛的网络应用之一. 针对 P2P 的研究很多, 其中研究的热点之一是负载均衡. 好的负载均衡算法能够降低 P2P 系统的通信开销, 提高系统的吞吐量和稳定性. 通常引起负载不平衡的原因是节点上的任务是随机到达的, 且任务完成时间不确定, 导致在运行过程中节点的负载是动态变化的. 因此, 在运行时动态测量并调整节点的负载, 可以大大提高系统的负载均衡度^[1-2].

现有的负载均衡算法均没有考虑节点对新到达请求的负载评估, 都是在节点负载超出阈值后, 才启动负载迁移算法, 属于被动的负载迁移, 负载迁移的代价增加了用户请求的响应时间, 降低了系统的服

务质量(Quality of Service, QoS).

本文提出了一种基于移动代理的主动负载均衡算法. 算法通过移动代理收集系统中各个节点的负载信息, 并产生负载迁移决策; 每个节点对新的用户请求, 通过评估其资源需求, 决定接受还是转移至其他节点, 以避免本节点过载^[3-4].

2 负载均衡算法

负载均衡算法的系统结构如图 1 所示. 系统中存在一个全局唯一的移动代理 MA (Mobile Agent), 它以轮询的方式在各个对等节点之间不停地移动, 采集每个节点的负载信息和处理能力, 形成一张系统负载信息表, 并据此作出负载迁移决策. 每个对等节点都独立地接受用户请求, 在每个节点上都驻留一个本地代理 LA (Local Agent), LA 保存一

份系统负载信息表的拷贝,该表在 MA 移动至该节点时被更新一次,LA 负责对新到达用户请求进行资源需求评估,如本节点不能满足需求,则根据系统负载信息表选择合适节点迁移用户请求。

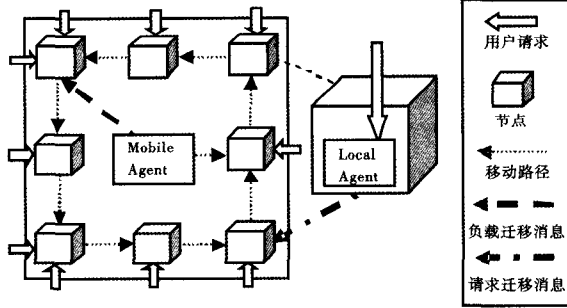


图1 系统结构图

负载均衡算法由三部分组成:负载信息收集、负载迁移、用户请求的评估与迁移。

2.1 负载信息收集

定义1 节点负载:用五元组 $\langle L_C, L_D, L_M, L_B, L_P \rangle$ 表示第*i*个节点 N_i 的负载,其中 L_C 表示CPU利用率, L_D 表示磁盘占用率, L_M 表示内存占用率, L_B 表示网络带宽占用率, L_P 表示进程数量占用率。由于P2P系统提供的服务类型可能不同,因此为每种参数引入一个权值 α_i , $\sum_{i=1}^5 \alpha_i = 1$,用来表示各项参数的重要程度。根据上述参数及权值,可得节点 N_i 的负载为:

$$Load_i = [L_C \ L_D \ L_M \ L_B \ L_P] \times [\alpha_1 \ \alpha_2 \ \alpha_3 \ \alpha_4 \ \alpha_5]'$$

定义2 节点处理能力:用五元组 $\langle C_C, C_D, C_M, C_B, C_P \rangle$ 表示第*i*个节点 N_i 的负载,其中 C_C 表示CPU处理速度, C_D 表示磁盘容量, C_M 表示内存容量, C_B 表示网络带宽, C_P 表示最大进程数量。由于P2P系统提供的服务类型可能不同,因此为每种参数引入一个权值 β_i , $\sum_{i=1}^5 \beta_i = 1$,用来表示各项参数的重要程度。根据上述参数及权值,可得节点 N_i 的处理能力为:

$$Capacity_i = [C_C \ C_D \ C_M \ C_B \ C_P] \times [\beta_1 \ \beta_2 \ \beta_3 \ \beta_4 \ \beta_5]'$$

定义3 系统负载信息表:假设一个P2P系统有*n*个节点,每个节点存储的负载信息用四元组 $\langle ID, Load, Capacity, Time \rangle$ 表示,其中ID是节点的全局唯一标识,Load存储节点的当前负载,Capacity存储节点的最大处理能力,Time存储信息采

集的时间戳。系统负载信息表*T*则表示为一个 $n \times 4$ 矩阵, $T_{i,j}$ 表示第*i*个节点的信息*j*。

移动代理MA以轮询的方式,周期性的遍历系统中的每个节点,采集该节点的负载信息,更新系统负载信息表,并将新表在本地生成一份拷贝,供本地代理LA使用。

由于P2P系统中^[5],用户请求具有随机性,如果用单次采样得到的瞬时负载来作为节点负载,则容易导致较大偏差,因此本算法使用简单的滤波方法,对采样数据进行平滑。设当前周期为*K*,负载采样值为 $L_{\text{samp}}(K)$,*K*-1周期的负载滤波值为 $L_{\text{filt}}(K-1)$,则*K*周期的负载滤波值 $L_{\text{filt}}(K) = L_{\text{filt}}(K-1) \times \alpha + L_{\text{samp}}(K) \times \beta$, $\alpha + \beta = 1$ 。参数 β 的大小决定了算法对瞬时负载的响应速度。

2.2 负载迁移

假设负载迁移的门限为 $Load_{\text{thred}}$,当MA收集到节点*i*的负载 $Load_i > Load_{\text{thred}}$ 时,认为该节点为重载节点,启动负载迁移算法。负载迁移算法的目的在于从所有轻载节点的集合 ϕ 中,寻找最优节点*j*作为迁移的对象,选择依据需综合考虑节点的负载余量和传输代价两种因素。

节点*i*需要迁移的负载量记作 $overLoad_i$, $overLoad_i = Load_i - Load_{\text{thred}}$;轻载节点*j*的负载余量记作 $spareLoad_j$, $spareLoad_j = Capacity_j - Load_j$;由节点*i*迁移到节点*j*的传输代价记作 $cost_{i,j}$,节点*j*的迁移权重的计算公式为: $Weight_j = spareLoad_j / cost_{i,j}$ 。

可以得出,节点*j*的负载余量越大,迁移权重越大;负载传输到节点*j*的代价越小,迁移权重越大。满足 $\text{MAX}(Weight_j)$, $j \in \phi$ 的节点*j*就是负载迁移的目的节点。

2.3 用户请求的评估与迁移

当新的用户请求到达时,为避免节点成为重载节点,由本地代理LA启动对该请求的资源需求评估。主要考虑以下几个关键性能参数为:CPU数量*C*,磁盘容量*D*,内存容量*M*,网络带宽数量*B*。根据经验性法则^[6]以及历史请求的先验知识,LA估算得到新请求的资源需求向量 $r = [R_C, R_D, R_M, R_B]$ 。收集本节点的当前负载 $l = [L_C, L_D, L_M, L_B]$,加上新请求带来的负载增量*r*,本节点的预测负载 $p = [P_C, P_D, P_M, P_B] = l + r$;使用定义1中的加权因子 α_i 对*p*进行加权,计算得到本节点的预测负载 $Load_{\text{pred}}$ 。若 $Load_{\text{pred}} < Load_{\text{thred}}$,说明该请求不会导致本节点成为重载节点,接受该请求;否则启动对该

请求的迁移算法.

LA 通过查询系统负载信息表,得到当前所有轻载节点的集合 ϕ . 与 2.2 节的负载迁移算法不同,对请求的迁移是由 LA 发起的,不同节点的请求迁移具有异步性和并发性,可能会出现多个请求迁移至同一轻载节点的情况,导致“颠簸”现象的产生. 本文假定不同用户请求之间的资源需求差异度较大,采用基于相似度的迁移算法,寻找与请求资源最接近的轻载节点,可有效解决“颠簸”问题.

轻载节点 j 的空闲负载 $s_j = Capacity_j - Load_j$, 定义资源的相似度函数为:

$$Sim(r, s) = \frac{\sum_{i=1}^n r_i \times s_i}{\sqrt{(\sum_{i=1}^n r_i^2) \times (\sum_{i=1}^n s_i^2)}}$$

其中, n 表示资源的种数, r_i 表示第 i 种资源的需求量, s_i 表示第 i 种资源的负载余量.

根据相似度函数计算每个轻载节点的相似度 $sim_j, j \in \phi$, 选择相似度最大的节点 j 作为请求迁移的目的节点.

3 仿真实验及分析

使用 PeerSim 作为模拟器,表 1 列出了实验环境和算法的相关参数. 假定 P2P 网络中有 1000 个节点,用户请求的资源需求服从 $[1, 20]$ 上的均匀分布.

表 1 实验参数设置

实验参数	值
网络节点个数	1000
用户请求的资源需求	$[1, 20]$
节点能力	$[50, 200]$

仿真实验从通信开销、服务质量和负载迁移量等三个方面分析负载均衡算法性能. 传统的分布式负载均衡算法称为 DLB (Distributed Load Balance), 本文的负载均衡算法称为 MAALB (Mobile Agent Based Active Load Balance).

实验结果表明: (1) 使用移动代理的负载均衡算法大大降低了通信开销, 减少了网络流量. (2) DLB 和 MAALB 算法的服务质量对比, 当用户请求频率较高时, MAALB 算法的用户响应速度更快, 服务质量更高. (3) 当重载节点比例较高时, MAALB 算法的负载迁移率要明显低于 DLB 算法.

4 结束语

本文提出了一种基于移动代理的负载均衡算法. 与传统的集中式和分布式负载均衡算法相比, 该算法使用移动代理收集负载信息、生成负载迁移决策, 大大降低了系统的通信开销, 而且避免了单点失效问题和“颠簸”现象. 通过引入对用户请求的评估和迁移, 预见性的处理负载问题, 提高了系统的服务质量. 在实际的 P2P 系统中, 用户请求虽然具有随机性, 但仍具有一定的时间相关性, 通过对节点历史负载数据的统计分析, 预测未来的负载情况, 将会进一步优化负载的分配, 提高系统的负载均衡程度, 这是下一步研究工作的重点.

参考文献:

- [1] Neeraj Nehra, Patel R B, Bhat V K. Load balancing in heterogeneous P2P systems using mobile agents[J]. World Academy of Science, Engineering and Technology, 2006;77-82.
- [2] Rupali Bhardwaj, Dixit V S, Anil Kr Upadhyay. A propound method for agent based dynamic load balancing algorithm for heterogeneous P2P systems[C]// International Conference on Intelligent Agent and Multi-Agent Systems. USA; Paul Przyborski, 2009.
- [3] Ezumalai R, Aghila G, Rajalakshmi R. Design and architecture for efficient load balancing with security using mobile agents[J]. IACSIT International Journal of Engineering and Technology, 2010, 2(1): 57-60.
- [4] Godfrey B Stoicaion. Heterogeneity and load balance in distributed hash tables[C]// Proceeding IEEE INFOCOM 2005. USA; Miami, 2005.
- [5] 陆垂伟, 李之棠, 林怀情, 等. 一种结构化 P2P 网络动态负载均衡算法的研究[J]. 计算机科学, 2008, 25(4): 1181-1183.
- [6] 余华山, 张宇, 吴宪国等. 一种基于运算量估算的任务群自适应计算技术[J], 高性能计算发展与应用, 2010, 4(33): 45-47.

作者简介:



李 慧 女, (1981—), 硕士, 讲师. 研究方向为对等计算、信息安全.