引用格式:孟晓景,张春勇. 一种改进加权最小连接算法及在 CDN 的负载均衡技术中的应用分析[J]. 山东科技大学学报(自然科学版), 2020, 39(1):85-90.

MENG Xiaojing,ZHANG Chunyong.An improved weighted least connection algorithm and it's application analysis in CDN load balancing technology[J]. Journal of Shandong University of Science and Technology (Natural Science),2020,39(1):85-90.

一种改进加权最小连接算法及在 CDN 的负载均衡技术中的应用分析

孟晓景,张春勇

(山东科技大学 计算机科学与工程学院,山东 青岛 266590)

摘 要:针对内容分发网络技术中本地负载均衡策略进行研究,提出一种适用于内容分发网络负载均衡的改进加权最小连接算法(improved weighted least connection,IWLC)。该算法在加权最小连接算法基础上考虑服务器性能与负载能力,引入服务器动态性能与负载水平评估因子,计算出综合性能指标以及动态权值并以此性能指标为依据完成任务调度,能较好地利用底层硬件资源,同时降低任务响应时间。

关键词:内容分发网络;动态负载均衡;改进加权最小连接;平均响应时间

中图分类号:TN929.5 **文献标志码**:A **文章编号**:1672-3767(2020)01-0085-06

DOI:10.16452/j.cnki.sdkjzk.2020.01.011

An improved weighted least connection algorithm and it's application analysis in CDN load balancing technology

MENG Xiaojing, ZHANG Chunyong

(College of Computer Science and Engineering,Shandong University of Science and Technology,Qingdao,Shandong 266590,China)

Abstract:This paper studies the local load balancing strategy in content distribution network technology and proposes an improved weighted leastconnection algorithm for content distribution network load balancing. The algorithm takes the server performance and load into consideration on the basis of the weighted minimum connection algorithm, introduces the server dynamic performance and load level evaluation factor, calculates the comprehensive performance index and completes the task scheduling based on the performance index to make good use of the underlying hardware resources and reducing theresponse time.

Key words:CDN;dynamic load balancing;an improved weighted least connection;average response time

内容分发网络[1]的内容路由技术一般由负载均衡系统来实现,是内容分发网络关键技术之一[2]。负载均衡的目的是运用多评估机制选择最佳节点并将用户路由到节点内合适的服务器上。内容路由决定了就近服务的思想能否实现,决定了整个内容分发网络系统的运行效率和性能,因此负载均衡是整个内容分发网络的基础[4]。常用的负载平衡算法有静态和动态两种。静态负载中,代表算法是轮询算法[5]和加权轮询算法[6]。静态负载平衡算法的优点是简单,缺点是由于不考虑服务器节点的性能和当前负载状态,长时间运行将导致负载不平衡。由于动态负载均衡算法[5]无需先验知识进行决策,可以根据实时服

收稿日期:2018-05-24

基金项目:赛尔网络下一代互联网技术创新项目(NGII20160205)

作者简介:孟晓景(1962—),男,浙江绍兴人,教授,硕士生导师,主要从事计算机网络的研究.E-mail:mxjzdy@163.com

务器负载状况,自适应调整任务在各服务器中执行顺序,从而提高了系统整体性能,因此在系统中得到广泛应用。现有的动态负载均衡算法主要有一致性哈希法[7]、最小连接算法(least connection, LC)[8]和加权最小连接算法(weighted least connection, WLC)[9]等。其中最小连接算法能动态获取服务器负载情况,但只关注连接数,没有关注 CPU 和内存使用率等问题,会出现连接数小但服务器资源消耗较高的情况[10-11];文献[12]提出一种异构集群负载索引和负载均衡算法,但对服务器负载情况的衡量只以任务数量为参考;文献[13]提出一种自适应权值最小负载实现负载均衡的算法,但没有考虑服务器所能承受的最大负载;文献[14]提出一种基于可变因子 α 的加权最小链接算法,根据各服务器的负载情况,对 α 的值进行调整并在一定程度上达到了负载均衡;文献[15]提出一种将终端连接数作为影响因子和各服务器参数进行综合分析的算法,没有考虑磁盘 I/O 且只应用于特定的场景中。上述文献对负载均衡算法都做了改进,但都有其片面性。本研究对最小连接算法进行改进得到一种 IWLC(improved weighted leastconnection)算法,提出了针对动态权值的关键参数新的计算方法。IWLC 算法通过动态获取服务器反馈性能指标与负载指标计算出当前集群中最优服务器进行任务的调度分配,可更好达到负载均衡效果,更好地解决 CDN 系统中大规模用户访问请求下的负载均衡问题。

1 算法介绍

1.1 典型负载均衡算法

1.1.1 最小连接算法

最小连接算法思想是:假设系统内各个服务器性能一致,当有新的任务到来时,总是将该任务分配给连接数最少的服务器[4,7],即对于现有服务器集群 $S=\{S_1,S_2,\cdots,S_n\}$,每个服务器连接数为 $C(S_i)$,$1\leqslant i<n$,当有新任务到达时,将其分配给连接数最少的服务器执行,假设 $C(S_k)=\min(C(Si))$,即分配给第 k 个服务器执行,同时更新该服务器连接数,$C(S_k)=C(S_k)+1$。LC 算法利用各服务器实时连接数来评价服务器负载情况,每次将任务分配给连接数最小的服务器执行,由于该算法没有考虑到各个服务器的性能指标,当各服务器性能相差较大时会影响负载均衡。

1.1.2 加权最小连接算法(IWLC)

加权最小连接算法算法[3]思想是在 LC 算法基础上,考虑了不同服务器的性能差异,设置不同的权值来标识服务器性能指标,在任务调度时为权值较大的服务器分配较大比例的活动连接[8,10]。IWLC 算法充分考量服务器性能指标及服务器动态负载指标,因此在负载均衡方面性能要优于最小连接算法;然而,现有的 IWLC 算法对性能指标的计算相对简单化,而系统实际运行过程中性能指标与负载指标更加复杂,因此 IWLC 算法仍有优化改进的空间。

1.2 改进加权最小连接算法(IWLC)

1.2.1 IWLC 算法思路及流程

CDN 应用系统中,负载均衡服务器实时接收各服务器节点的负载情况及资源使用状态,IWLC 算法利用实时获取的参数来计算各服务器节点的综合性能指标并以此作为任务调度的依据,选择出最适合执行任务的服务器,从而完成任务的动态调度。

IWLC 算法流程如下:

1) 集群中各真实服务器节点周期性计算各自的 CPU 空闲率、内存空闲率、磁盘 I/O 空闲负荷、负载连接数,并反馈给负载均衡调度器;

2) 负载均衡调度器根据各服务器节点反馈的信息,计算服务器的综合性能指标 $P(S_i)$、综合负载指标 $C(S_i)$ 以及综合评价指标参数 $F(S_i)$,并计算出各个结点的权值 w_i;

3) 负载均衡调度器根据计算出的权值修改原先的权重,将更新后的权重写入 IPVS 内核;

4) Linux 内核启用 LVS 编译安装 IPVS 内核,IWLC 算法根据新的权值选择最合适的服务器进行任务分配。

1.2.2 IWLC算法中关键参数计算

1) 集群S由n台服务器组成,$S=\{S_1,S_2,\cdots,S_n\}$;

2) 假设C_i、M_i、D_i、N_i、L_i分别为服务器S_i的CPU空闲率、内存空闲率、磁盘I/O空闲负荷、网络带宽、负载连接数;$R_{\text{cpu}}^{(i)}$、$R_{\text{mem}}^{(i)}$、$R_{\text{disk}}^{(i)}$、$R_{\text{net}}^{(i)}$、$R_{\text{con}}^{(i)}$分别为服务器S_i的CPU空闲率比值、内存空闲率比值、磁盘I/O空闲负荷比值、网络带宽比值、负载连接数比值。服务器S_i的$R_{\text{cpu}}^{(i)}$、$R_{\text{mem}}^{(i)}$、$R_{\text{disk}}^{(i)}$、$R_{\text{net}}^{(i)}$、$R_{\text{con}}^{(i)}$的计算方法如下:

$$R_{\text{cpu}}^{(i)}=\frac{C_i}{\max(C_1,C_2,\cdots,C_n)},1\leqslant i\leqslant n\ , \tag{1}$$

$$R_{\text{mem}}^{(i)}=\frac{M_i}{\max(M_1,M_2,\cdots,M_n)},1\leqslant i\leqslant n\ , \tag{2}$$

$$R_{\text{disk}}^{(i)}=\frac{D_i}{\max(D_1,D_2,\cdots,D_n)},1\leqslant i\leqslant n\ , \tag{3}$$

$$R_{\text{net}}^{(i)}=\frac{N_i}{\max(N_1,N_2,\cdots,N_n)},1\leqslant i\leqslant n\ , \tag{4}$$

$$R_{\text{con}}^{(i)}=\frac{L_i}{L_{\text{max}}^{(i)}}\ 。 \tag{5}$$

其中:$\max(C_1,C_2,\cdots,C_n)$为当前所有服务器中CPU空闲率的最大值;$\max(M_1,M_2,\cdots,M_n)$为当前所有服务器中内存空闲率的最大值;$\max(D_1,D_2,\cdots,D_n)$为当前所有服务器中磁盘I/O空闲负荷最大值;$\max(N_1,N_2,\cdots,N_n)$为当前所有服务器中网络带宽的最大值;$L_{\text{max}}^{(i)}$为当前服务器所能负载连接数的最大值。负载均衡调节器选择CPU空闲率、内存空闲率、磁盘I/O空闲负荷、网络带宽均大于90%的服务器来进一步计算,而那些不符合该条件的服务器权值设为0。

3) 用$P(S_i)$和$C(S_i)$分别表示服务器S_i的综合性能指标与综合负载指标。定义$C(S_i)=R_{\text{con}}^{(i)}$;$P(S_i)$由该服务器的CPU空闲率比值和内存空闲率比值、磁盘I/O空闲负荷比值和网络带宽比值等4个性能因素指标计算得到,方法如下:

$$\begin{cases}P(S_i)=w_1*R_{\text{cpu}}^{(i)}+w_2*R_{\text{mem}}^{(i)}+w_3*R_{\text{disk}}^{(i)}+w_4*R_{\text{net}}^{(i)}\\ w_1+w_2+w_3+w_4=1\end{cases}, \tag{6}$$

其中w_1、w_2、w_3、w_4分别代表各服务器性能指标因素的权重。考虑到CPU空闲率和内存空闲率两个指标在描述服务器性能方面的重要性,因此对于4个权重的赋值分别是:$w_1=0.4$、$w_2=0.4$、$w_3=0.1$、$w_4=0.1$。若当前的系数w_1、w_2、w_3、w_4不能很好反映应用的负载,可以对系数进行不断的修改,直到找到一组贴近当前应用的系数。经过多次试验,判定系数$w_1=0.4$、$w_2=0.4$、$w_3=0.1$、$w_4=0.1$是最优的。

4) 为了更加有效预测每台服务器节点所能承受的负载能力,引入服务器的综合评价指标参数$F(S_i)$,其计算方法如下:

$$F(S_i)=\frac{C(S_i)}{P(S_i)},\quad 1\leqslant i\leqslant n。 \tag{7}$$

$F(S_i)$的变化规则为:若要使$F(S_i)$取值最小,其对应的服务器S_i要么满足服务器性能指标$P(S_i)$最大,要么满足服务器负载指标$C(S_i)$最小,或者同时满足这两个条件,此时S_i便是集群中最合理的执行新任务的服务器。

5) 设服务器的S_i权重为w_i,则令$w_i=1/F(S_i)$。$F(S_i)$取值越小,服务器S_i的动态权重取值越大,负载指标就越小,服务器S_i的连接数就越小。

1.2.3 IWLC算法的代码及分析

IWLC算法只做一次for循环,依次比较各服务器综合指标参数$F(S_i)$,$F(S_i)$取值大的被舍弃,继续比较下一个,直到第n个服务器,选出$F(S_i)$值最小的作为下一轮要执行新任务的服务器,算法时间复杂度为$O(n)$,可见对参数的改进并没有造成更高的时间复杂度。

IWLC 算法通过本研究提出的动态权值计算方法得到服务器综合评价指标值,将新任务分配到 $F(S_i)$ 值最小的服务器 S_i 上执行,此时 S_i 是当前集群中最适合承担负载的服务器。

Algorithm: Improved Weighted Least Connection

Input: CPU idle rate C_i, Memory idle rate M_i, I/O rate D_i, Bandwidth N_i, Number of load connections L_i Number of server n, Weight w_1, w_2, w_3, w_4

Output: *The most weighted server* S_i

Initialize an Array of n-dimensions arr with 0

for($i = 1; i \leqslant n; i++$)

do

$R_{\text{cpu}}^{(i)} = C_i / \max(C_1, C_2, \cdots, C_n)$

$R_{\text{mem}}^{(i)} = M_i / \max(M_1, M_2, \cdots, M_n)$

$R_{\text{disk}}^{(i)} = D_i / \max(D_1, D_2, \cdots, D_n)$

$R_{\text{net}}^{(i)} = N_i / \max(N_1, N_2, \cdots, N_n)$

$R_{\text{con}}^{(i)} = \dfrac{L_i}{L_{\max}^{(i)}}$, $C(S_i) = R_{\text{con}}^{(i)}$

$P(S_i) = w_1 * R_{\text{cpu}}^{(i)} + w_2 * R_{\text{mem}}^{(i)} + w_3 * R_{\text{disk}}^{(i)} + w_4 * R_{\text{net}}^{(i)}$

$F(S_i) = C(S_i) / P(S_i)$, $W_i = 1 / F(S_i)$

$W = W_i$

$arr[i] = W$

Write the weight W_i to the IPVS kernel and make install the IPVS kernel

end do

$W_{\max} = arr[i]_{\max}$

return the most weighted server S_i

2 实验与分析

2.1 实验设计

实验模拟 CDN 应用系统中边缘服务器节点应对客户访问请求的负载均衡效果,本研究利用实验室现有设备,基于章文嵩博士[9]所提出的 Linux 虚拟服务器开源项目的集群体系结构技术搭建了一套模拟 CDN 应用系统边缘服务器节点系统。

为验证提出的 IWLC 算法在解决 CDN 应用系统中集群负载均衡问题方面的有效性,同时对比本算法与加权最小连接算法[5]在服务器响应时间与吞吐量两个评价指标上的实现效果,设计具体实验如下:

1) 在负载均衡服务器上通过 IPVSADM 管理软件部署本研究提出的 IWLC 算法,而 IPVSADM 自身已经配置了常用负载均衡算法,如加权轮询算法加权最小连接算法等;

2) 在测试客户机器上安装部署 LoadRunner 系统负载测试工具,用来记录模拟用户访问请求后台服务器内容过程中系统平均响应时间与吞吐量;

3) 分 10 组模拟用户访问过程,请求任务数以每组递增 100 的方式从 100 到 1 000 进行实验,利用 LoadRunner 记录系统平均响应时间与吞吐量指标,每组测三次,最终取平均值;

4) 用 MATLAB 画图对比本研究所提 IWLC 算法与加权最小连接算法的系统平均响应时间和系统吞吐量。

2.2 实验结果分析

提出的 IWLC 算法与 WLC 算法的系统平均响应时间对比实验结果如表 1 和图 1 所示。可以看出:当任务请求数低于 300 时,IWLC 算法对应的系统平均响应时间要长于 WLC 算法,原因在于 IWLC 算法需要实时获取各服务器传送过来的性能指标及负载指标以计算各服务器的综合性能指标,引起额外的资源消耗,在任务请求数量较低的情况下造成了系统响应时间延长;但是随着系统用户访问请求任务数的增加,由于 IWLC 算法能够充分利用系统资源,有效调度用户任务请求,IWLC 算法优于 WLC 算法。

表 1 10 组平均响应时间对比实验结果表

Tab.1 Experimental results of average response time

实验组	任务请求数	平均响应时间/ms	
		本文提出算法	加权最小连接算法
1	100	366	336
2	200	438	408
3	300	460	453
4	400	474	499
5	500	442	522
6	600	458	528
7	700	504	534
8	800	515	553
9	900	571	589
10	1 000	579	646

2) IWLC 算法与 WLC 算法的系统吞吐量对比实验如图 2 所示。由图可见:当任务数小于 500 时,IWLC 算法测试出的系统吞吐量小于加权最小连接算法;当任务数为 600～1 000 时,IWLC 算法测试出的系统吞吐量逐渐超过 WLC 算法测试出的系统吞吐量,大约提高了 1 347 bytes/ms,随着任务请求数的增多,IWLC 算法对于合理使用各服务器更加有优势。

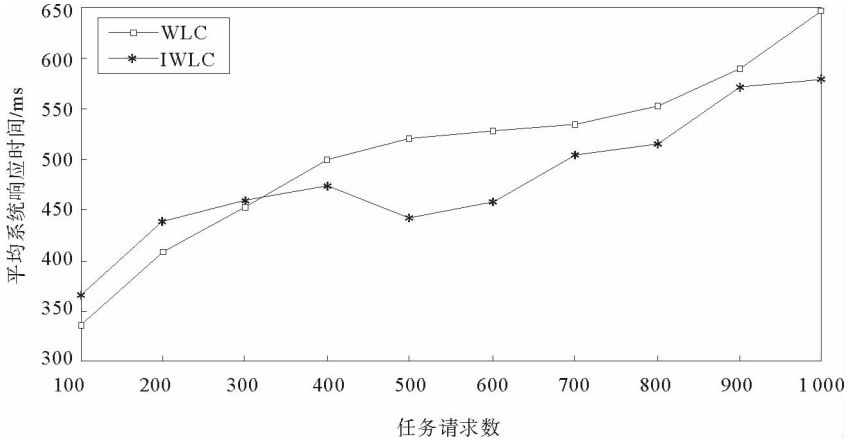

图 1　平均响应时间实验结果对比图

Fig. 1　Comparison of Experimental Results of Average Response Time

综上,IWLC 算法有效提高了系统平均吞吐量,尽可能避免服务器过载,将任务分配到具有最佳性能的服务器,加强系统资源的有效利用。

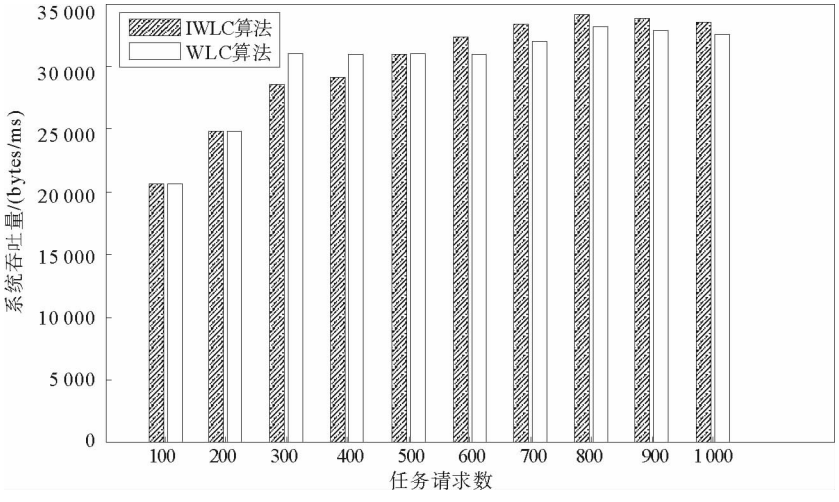

图 2　系统吞吐量实验结果对比图

Fig. 2　Comparison of experimental results of throughput

3　结论

本研究从 CDN 应用系统的本地负载均衡方面,针对边缘服务器如何应对大量用户访问请求而保障负载均衡的问题,基于加权最小连接算法,提出了一种改进加权最小连接的 CDN 负载均衡算法。

相比于 WLC 算法,IWLC 算法充分考虑了服务器自身资源使用情况以及负载情况等因素,通过动态实时收集各服务器节点的资源性能指标及负载指标,综合计算出当前的最优节点,完成访问请求任务的分配,保证最大限度地利用系统资源,并对用户的访问请求更快地做出响应。但 IWLC 算法也存在不足,如任务请求数较少时,负载均衡调节器定时发送请求报文获取并处理真实服务器的各种信息带来的额外开销会影响整个系统的测试与评估。本工作没有考虑到当 CPU 空闲率、内存空闲率、磁盘 I/O 空闲负荷、网络带宽四个参数都不大于 90%的服务器时的算法情况,比较理想化,需要进一步研究和细化算法来解决上述情况下的问题。

参考文献:

[1]RAJKUMAR B,MUKADDIM P T,ATHENA V K.Content Delivery Networks [M].Berlin : Springer-Verlag Berlin Heidelberg,2010:3-197,275-316.

[2]PATHAN A M K,BUYYA R.A taxonomy and survey of content delivery networks[J].Grid Computing and Distributed

Systems Laboratory,University of Melbourne,Technical Report,2007.

[3]张玉洁,孟祥武.基于用户需求的内容分发点对点网络系统研究[J].软件学报,2014,25(1):98-117.
Zhang Yujie,Meng Xiangwu. Research on CDN-P2P System over User Requirements[J].Journal of Software,2014,25(1):98-117.

[4]HE H,FENG Y,LI Z,et al.Dynamic load balancing technology for cloud-oriented CDN[J].Computer Science and Information Systems,2015,12(2): 765-786.

[5]MHARCHOL-BALTER M,DOWNEY A B.Exploiting process lifetime distributions for dynamic load balancing [J].ACM Transactions on Computer Systems,1997,15(3):253-285.

[6]SSHARIFIAN S,METAMEDI S A,AKBARI M K.An approximation-based load-balancing algorithm with admissioncontrol for cluster Web servers with dynamic workloads[J].The Journal of Supercomputing,2010,53(3):440-463.

[7] KARGER D,LEHMAN E,LEIGHTON T,et al. Consistent hashing and random trees: Distributed caching protocols for relieving hot spots on the World Wide Web.[C]// Twenty-ninth Acm Symposium on Theory of Computing. 1997.

[8]GENOVA Z,CHRISTENSEN K J.Challenges in URL switching for implementing globally distributed Web sites[C]// International Workshops on Parallel Processing.IEEE, 2000:89-94.

[9]AKON M,DRUSCHEL P,ZWAENEPOEL W.Efficieat support for PHTTP in cluster-based web servers[C]// USENIZ AMUAL Technical Conference,General Track,1999:185-198.

[10]包晓安,魏雪,陈磊,等.基于 mean-variance 的服务集群负载均衡方法[J]. 电信科学, 2017(1):1-8.
BAO Xiaoan,WEI Xue,CHEN L,et al. Load balancing method of service cluster based on mean-variance[J].Telecommunications Science, 2017(1):1-8.

[11]刘敏,王红斌.一种新的处理能力优先的权值分配调度算法[J].吉林大学学报(理学版),2011,49(6):1105-1109.
LIU Min,WANG Hongbin.A new type weights distribution scheduling algorithm with priority processing capability[J]. Journal of Jilin University(Science Edition),2011,49(6):1105-1109.

[12]BOSQUE J,TOHRIA P,ROBLES D,et al.ALoad index and load balancing algorithm for heterogeneous clusters[J].The Journal of Supercomputing,2013,65(3):1104-1113.

[13]杨婷,万良.一种自适应权值最小负载的 LVS 集群负载均衡算法[J].通信技术,2017(4):741-745.
YANG Ting,WAN Liang. LVS cluster load balancing algorithm with adaptive weight leastload [J]. Communications Technology, 2017(4):741-745.

[14]朱振广.面向内容分发网络的动态负载均衡技术研究[D].哈尔滨:哈尔滨工业大学,2012.
ZHU Zhenguang. Research on technology of dynamic load balancing for CDN[D]. Harbin: Harbin Institute of Technology,2012.

[15]高振斌,潘亚辰.改进的基于加权最小链接数的负载均衡算法[J].科学技术与工程,2016(6):81-85.
GAO Zhenbin,PAN Yachen.Improved load balancing algorithm based on weighted least-connections[J]. Science Technology and Engineering,2016(6):81-85.

[16]SIAVOSHANI M J, SHARIATPANAHI S P, GHASEMI H,et al. On communication cost vs. load balancing in Content Delivery Networks[C]// IEEE Symposium on Computers and Communications (ISCC). 2017: 651-656.

[17]李乔,何慧,张宏莉.内容分发网络研究[J].电子学报,2013,41(8):1560-1568.
LI Qiao,HE Hui,ZHANG Hongli.Research on content delivery networks[J]. Acta Electronica Sinica,2013,41(8):1560-1568.

[18]章文嵩.可伸缩网络服务的研究与实现[D].长沙:国防科学技术大学,2000.
ZHANG Wensong.The study and implementaion of Scalbale network sevrices[D].Changsha: National University of Defense Technology,2000.

[19]张玉芳,魏钦磊,赵膺.基于负载权值的负载均衡算法[J].计算机应用研究,2012,29(12):4711-4713.
ZHANG Yufang,WEI Qinlei,ZHAO Ying.Load balancing algorithm based on load weights[J]. Application Research of Computers, 2012,29(12):4711-4713.

(责任编辑:傅 游)