文章编号:1671-4598(2016)07-0219-05 DOI:10.16526/j.cnki.11—4762/tp.2016.07.059 中图分类号:TP301.6 文献标识码:A

面向 SLA 的云计算负载均衡策略

雷显玉，刘 勇，黄广君
(河南科技大学 信息工程学院，河南 洛阳 471023)

摘要：云计算负载均衡是保障 SLA 协议的关键问题之一；针对云计算负载均衡问题，提出一种面向 SLA 的负载均衡策略；该策略引入人工神经网络思想，建立负载均衡模型，采用单层感知器算法（SLPA）将虚拟机负载状态进行分类，然后利用结合了动态加权轮询算法的 BP 神经网络算法（BPNNA—DWRRA）有针对性地对虚拟机负载权重进行预测更新，最后将任务调度到最小权重所对应的可行虚拟机上；应用 CloudSim 进行仿真实验，结果表明了该策略的可行性，同时，相比加权最小链接算法和粒子群算法，该策略的平均响应时间分别节省了 43.6%和 22.5%，SLA 违反率分别降低了 20.7%和 14.4%；因此，所提策略在响应用户任务时，请求响应时间短，SLA 违反率低，保障了 SLA。

关键词：服务等级协议；云计算；负载均衡；人工神经网络；动态加权轮询

Load Balancing Strategy of Cloud Computing Oriented to SLA

Lei Xianyu, Liu Yong, Huang Guangjun
(School of Information Engineering, Henan University of Science and Technology, Luoyang 471023, China)

Abstract: Load balancing of the cloud computing is one of the key issues of guaranteeing SLA agreement. Load Balancing for cloud computing, a load balancing strategy oriented SLA is put forward. Introduced artificial neural network thinking, this strategy establishes load balancing model. VMs load status is classified by single—layer Perceptron algorithm (SLPA), then BP neural network algorithm (BPNNA) combined with dynamic weighted round—robin algorithm (DWRRA) targetedly predicts and updates VMs load weight, finally, the task is dispatched to the feasible virtual machine according to the minimum weight. The CloudSim simulation experiment was performed, and the results show that the proposed strategy is feasible. Meanwhile, compared with the Weighted Least—Connection algorithm and particle swarm optimization algorithm, the strategy of this paper respectively reduce average response time by 43.6% and 22.5%, and reduce SLA violation rates by 20.7% and 14.4%. Therefore, the proposed strategy in response to user tasks, make response time short, SLA violation rate low, and SLA is guaranteed.

Keywords: service level agreement; cloud computing; load balancing; artificial neural network; dynamic weighted round—robin

0 引言

云计算是一种商业服务模式，云服务提供商为用户提供服务，用户按需使用并按量付费，同时，双方签订 SLA[1] (service level agreement，服务等级协议）来确保自身利益和服务质量。但是云计算环境下服务器之间存在的异构性导致在资源分配的过程中会产生负载不均衡[2]，使集群整体性能下降，用户服务质量无法得到保障，云服务提供商也需要因 SLA 违约而缴纳罚金。因此，如何均衡系统负载，减少 SLA 违反率，保障服务质量成为亟待解决的问题。

为解决负载均衡问题，国内外研究者已经提出了多种不同的算法和策略，文献[3]设计了一种改进的基于动态反馈机制的负载均衡算法，该算法实现了有效的动态负载平衡，简化了负载均衡器的任务分配算法。文献[4]是基于模拟退火算法和动态加权轮询的负载均衡策略，该策略能自适应并且精确地确定性能权重向量，实时计算应用服务器负载，动态分配请求。文献[5]提出了一种基于动态优先级和萤火虫行为的云任务调度算法，根据任务动态优先级和决策变量将任务调度到合适的虚拟机上。文献[6]提出了一种改进的加权最小连接数负载均衡调度算法，即建立对节点信息的实时收集以及负载量变化的模型，该模型按照负载情况来合理有效分配资源。文献[7]提出了一种综合性的资源分配策略，其中包括一个数据密集型任务调度算法和虚拟机分配策略，保持物理主机的良好负载均衡的同时还可以大大减少任务完成时间。文献[8]结合加权最小链接算法和粒子群算法来解决虚拟机调度问题，利用历史信息来预测新的输入请求的工作量，通过拒绝无法调度来减少调度过程的计算时间，该算法可以保持负载均衡。文献[9]考虑到每个虚拟机的负载和计算能力，同时引入蚂蚁的向前移动和向后移动的思想，处理了云环境中的复杂的任务调度，缩短了总任务的完成时间。文献[10]建立资源—任务分配模型，分别运用离散粒子群算法实现资源负载均衡。文献[11]在保证用户服务质量的前提下，提出了一种新的负载平衡机制，考虑到云供应商的同时还考虑了用户的服务质量，设计任务调度算法和弹性伸缩算法来实现任务分配。

收稿日期:2015-12-30； **修回日期**:2016-01-29。

作者简介:雷显玉(1989-)，女，河南信阳人，硕士研究生，主要从事云计算方向的研究。

刘 勇(1966-)，男，湖南岳阳人，教授，硕士研究生导师，主要从事云计算、软件体系结构方向的研究。

黄广君(1963-)，男，河南洛阳人，教授，硕士研究生导师，主要从事语义网、自然语言处理方向的研究。

上述工作研究实现了有效的负载平衡，但是忽略了从保障SLA的角度出发，评判负载均衡算法是否可以满足用户需求，保障用户自身利益。为解决这些问题，本文提出了一种面向SLA的负载均衡策略。

1 面向 SLA 的负载均衡模型

在云计算环境中，有效的负载均衡算法可以提高服务质量，而SLA是衡量服务质量的主要依据。本文负载均衡策略建立了面向SLA的负载均衡模型，模型如图1所示。

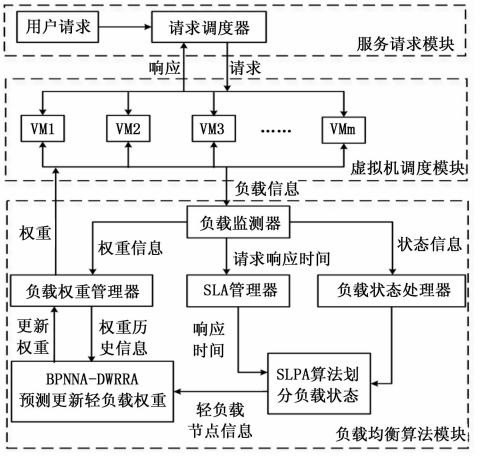

图1 负载均衡模型

该模型由三大模块组成，分别是服务请求模块、虚拟机调度模块和负载均衡算法模块。

1）服务请求模块：

负责接收服务请求，并将请求转发给请求调度器。

2）虚拟机调度模块：

从请求调度器得到服务请求，根据负载均衡算法模块为请求找到合适的虚拟机节点，并响应用户请求。

3）负载均衡算法模块：

此模块是基于人工神经网络的负载均衡算法，主要负责为虚拟机调度模块提供合适的虚拟机节点，其中包括6个子模块，分别是负载监测器模块、负载状态管理器模块、负载权重管理器模块、SLA管理器模块、SLPA算法划分负载状态模块（简称SPLA算法模块）和BPNNA－DWRRA算法预测更新轻负载权重模块（简称BPNNA－DWRRA算法模块）。负载监测器负责监测所有虚拟机的负载状态、负载权重信息和对服务请求的响应时间，负载状态包括CPU使用率、内存使用率，硬盘使用率和网络带宽使用率；负载状态管理器负责管理负载的当前状态信息和历史状态信息；负载权重管理器负责管理负载的当前权重信息和历史权重信息；SLA管理器负责管理请求响应时间信息，其包括SLA中规定的请求平均响应时间和服务请求响应时间；SPLA算法模块和BPNNA－DWRRA算法模块都是基于人工神经网络的，SLPA算法模块根据负载状态管理器提供的负载信息和SLA管理器提供的响应时间信息将所有负载进行类型划分，划分为重载节点和轻载节点；根据SLPA算法得到的轻载节点，负载权重管理模块为BPNNA－DWRRA算法模块提供所有轻载节点的权重历史信息，然后BPNNA－DWRRA算法将所有轻载节点的权重进行预测更新，并将更新的权重交给负载权重管理器进行记录，最后负载权重管理器中节点权重最小的即为合适的虚拟机节点。

2 基于人工神经网络的负载均衡算法

为使用户找到合适的虚拟机节点来响应服务请求，本文算法利用人工神经网络的泛化能力[12]，将实现分为两个阶段，先将系统中所有虚拟机节点进行状态划分再根据分类信息对虚拟机节点权重进行预测更新，即负载状态分类、负载权重预测更新，最终利用虚拟机节点权重来动态分配请求任务。

服务器系统中有 m 台虚拟机资源节点，定义虚拟机资源节点的集合是 $H=H_i=[H_1, H_2, H_3, \cdots, H_m]$（$i=1, 2, \cdots, m$），并用 CPU 使用率、内存使用率，硬盘使用率和网络带宽使用率作为评价虚拟机节点的性能指标，性能指标对应的描述为：H^i_{CPU}，H^i_{mem}，H^i_{dio}，H^i_{net}，其各自对应的权值为 r_c，r_m，r_d，r_n。通过负载状态管理器提供的所有负载的性能状态，来计算虚拟机节点 i 当前负载，虚拟机节点 i 当前负载 $Load$（H_i）用公式（1）表示。

$$Load(H_i)=L_i=[r_c,r_m,r_d,r_n]\begin{bmatrix}H^i_{CPU}\\H^i_{mem}\\H^i_{dio}\\H^i_{net}\end{bmatrix} i\in[1,m] \quad (1)$$

公式（1）其中：$0<L_i<1, r_c+r_m+r_d+r_n=1$

2.1 负载状态分类

单层感知器模拟的是人工神经网络，它属于单层前向网络，即除了输入层和输出层之外只拥有一层神经元节点，是最简单的神经网络，用来解决线性可分的二分类问题[13]。本文利用单层感知器算法SLPA对所有虚拟机节点进行智能分类，将虚拟机节点划分为轻载节点和重载节点。SLPA算法模型如图2所示。

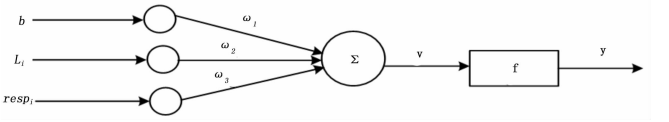

图2 SLPA算法模型

定义1：SLPA算法模型。

SLPA算法模型定义为一个四元组：

$$Model=(X,\omega,v,y)$$

$X=X(n)$ 表示SLPA算法模型的第 n 次迭代的输入向量，$X=[b,L_i,resp_i]$，其中 b 表示感知器的输入偏置，设为固定值1，即 $b=1$；L_i 是由公式（1）计算得到的服务虚拟机节点 i 的当前负载；$resp_i$ 是由SLA管理器提供的虚拟机节点 i 的请求响应时间。$\omega=\omega(n)$ 表示第 n 次迭代的连接权值向量，$\omega=[\omega_1,\omega_2,\omega_3]$。$v=v(n)$ 表示第 n 次迭代的输出，$Y=Y(n)$ 表示第 n 次迭代的实际输出，分别用由公式(2)(3)表示，其中公式(3) f 是SLPA算法的传递函数。

$$v=\sum_{j=1}^{3}x_j\omega_j=X(n)\omega(n)(j=1,2,3) \quad (2)$$

$$y=f(v) \quad (3)$$

SLPA 算法描述如下：

1）定义变量 η 为学习率，ε 为误差极限，迭代次数 n。

2）初始化权值向量 ω 和变量 η，ε，n。

3）初始化输入样本。

利用负载状态管理器采集所有虚拟机节点的历史负载状态信息，通过公式（1）计算得到虚拟机节点 i 历史当前负载，同时，利用 SLA 管理器提供虚拟机节点 i 的历史请求响应时间，生成输入向量 X_i。d_i 表示输入向量 X_i 的期望输出，其值代表了响应服务的虚拟机节点 i 是否满足了 SLA 协议，0 表示违反了 SLA 协议，1 表示未违反 SLA 协议，d_i 用公式（4）表示。用 P 表示输入样本，指定 P 为 k 维，其由输入向量 X 和 d 组成，P 用公式（5）表示。

$$d_i = \begin{cases} 0 & resp_i \geqslant resp_{SLA} \times 80\% \\ 1 & resp_i < resp_{SLA} \times 80\% \end{cases} \tag{4}$$

$$p = \begin{bmatrix} X_1 & X_2 & \dots & X_k \\ d_1 & d_2 & \dots & d_k \end{bmatrix} \tag{5}$$

4）训练网络，计算实际输出 y。

输入样本 p，根据公式（2）（3）得到第 n 次迭代的实际输出 y（n）。

5）判断算法是否收敛。

输出误差 ε 即为期望输出和实际输出之差，用公式（6）表示，若计算得到的 ε 值小于初始化的 ε 值时，算法收敛，该算法结束；若不满足，利用公式（7）来更新权值向量 ω，且循环次数增加 1，转到第 3 继续执行。

$$\varepsilon = d - y \tag{6}$$

$$\omega(n+1) = \omega(n) + \eta(d(n) - y(n))L(n) \tag{7}$$

2.2　负载权重预测更新

SLPA 算法将虚拟机节点划分为轻载节点和重载节点，在 BPNNA－DWRRA 算法对负载权重预测更新时，只对轻载节点的权重进行预测更新，缩小虚拟机节点权重预测更新范围，提高效率。

BPNNA－DWRRA 算法是基于 BP 神经网络算法和动态加权轮询算法的混合算法，BP 神经网络包含了多个隐含层，是一个层内无相互连接结构的、无反馈的前向网络，其收敛速度慢，存在局部极小点等[14]，但是本文 BPNNA－DWRRA 算法能自适应地调整学习率和增加动量项，改善了 BP 神经网络收敛问题，其模型如图 3 所示。

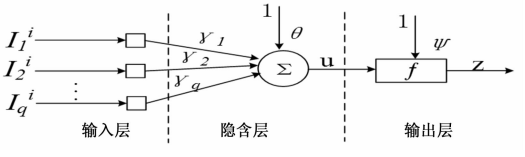

图 3　BPNNA－DWRRA 算法模型

定义 2：BPNNA－DWRRA 算法模型。

BPNNA－DWRRA 算法模型定义为一个六元组：

$$BPModel = (I, \gamma, \theta, u, \varphi, z) \tag{8}$$

$I=I$（e）表示虚拟机节点的历史权重集合，e 表示当前的迭代次数，$I=I_i=[I_1^i, I_2^i, \cdots, I_q^i]$，其中 $Ii\ q$ 是由权重管理器提供的虚拟机节点 i 的 q 维权重历史集合。$\gamma=\gamma$（e）表示神经网络第 e 次迭代的输入层至隐含层的连接权值，$\gamma=[\gamma_1, \gamma_2, \cdots, \gamma_q]$，$\theta$ 为隐含层的阈值，u 表示虚拟机节点 i 的隐含层至输出层的连接权值，φ 为输出层的阈值，z 表示输出层的实际输出。

BPNNA－DWRRA 算法描述如下：

1）定义并初始化误差容限 δ，学习率 σ，动量因子 λ，迭代次数 e。

2）指定输入样本。

依据 SLPA 算法得到的重载节点信息，从权重管理器中得到所有重载节点的权重历史数据 data，指定 data 为 t 维，用公式（9）表示。用 out_i 表示虚拟机节点 i 的期望输出，其值是虚拟机节点 i 的输出权重，由公式（10）表示。

$$data = [I_1, I_2, \cdots, I_t] = \begin{bmatrix} I_1^1 & I_2^1 & \cdots & I_t^1 \\ I_1^2 & I_2^2 & \cdots & I_t^2 \\ \vdots & \vdots & & \vdots \\ I_1^q & I_2^q & \cdots & I_t^q \end{bmatrix} \tag{9}$$

$$out_i = \frac{L_i}{\sum_{g=1}^{q} I_g^i} \tag{10}$$

公式（10）中，L_i 是通过公式（1）得到的虚拟机节点 i 的当前负载，$\sum_{g=1}^{q} I_g^i$ 是虚拟机节点 i 的 q 维历史权重之和。指定 inData 为输入样本，其由 data 和 out 组成，用公式（11）表示。

$$inData = \begin{bmatrix} I_1 & I_2 & \dots & I_t \\ out_1 & out_2 & \dots & out_t \end{bmatrix} \tag{11}$$

3）训练网络。

对 $inData$ 输入样本进行划分，划分为训练数据 $trainData$ 和测试数据 $testData$，训练数据用来训练 $BPNNA-DWRRA$ 算法，测试数据用来测 $BPNNA-DWRRA$ 算法的准确性，$trainData$ 和 $testData$ 分别用公式（12）和（13）表示，其中 td 表示训练数据 $trainData$ 的维数。

$$trainData = \begin{bmatrix} I_1 & I_2 & \dots & I_{td} \\ out_1 & out_2 & \dots & out_{td} \end{bmatrix} \tag{12}$$

$$testData = \begin{bmatrix} I_{td+1} & I_{td+2} & \dots & I_t \\ out_{td+1} & out_{td+2} & \dots & out_t \end{bmatrix} \tag{13}$$

输入训练数据 $trainData$，训练网络，得到实际输出 z 用公式（14）表示，其中 f 是网络传递函数。

$$z = f(f(\sum_{t=1}^{q} I_t\gamma_t + \theta)\mu + \varphi) = f(f(\gamma(e)I(e) + \theta)\mu + \varphi) \tag{14}$$

4）判断算法是否收敛。

输出误差 ζ 用公式（15）表示，若误差 ζ 的平方和小于误差容限 δ 时算法收敛，直接进行下一步的网络测试；若不满足，进行误差反向传播，采用动态加权轮询算法对权值进行更新调整，权值更新用公式（16）表示。

$$\xi = out(n) - z(n) \tag{15}$$

$$\gamma_t = \gamma_{t-1} + (1-\lambda)\sigma\zeta + \lambda\psi \tag{16}$$

其中：ζ 权值修改量，ψ 上一次的权值修改量。权值修改量的值是网络层之间的局部梯度和网络输出的乘积，根据公式（16）对网络连接权值进行更新之后，转到第 3 步继续执行。

5）网络测试。

输入测试数据 *testData*，执行步骤 3）和 4）得到期望输出。

网络训练和测试完毕之后，依据 SLPA 算法得到轻载节点信息，从权重管理器中得到所有轻载节点的权重历史数据，然后将得到的数据输入网络，对所有轻载节点进行权值预测，并将预测后的虚拟机节点权重存在权重管理器中，最终虚拟机调度模块将任务调度到权重最小的虚拟机上，响应服务请求。

3 SLA 违反率

基于人工神经网络的负载均衡算法分别利用 SLPA 算法和 BPNNA－DWRRA 算法实现了为用户找合适的虚拟机资源，并响应用户请求。但是，该算法是否能够满足用户可接受的服务质量，还是需要作出有力的判断。本文使用 SLA 违反率作为其判断的依据，SLA 违反率定义为虚拟机上用户请求的违反次数除以用户请求的总次数。即在相同请求次数的情况下，请求违反次数越多，SLA 违反率越高，服务质量越差，算法效率越低。

定义 3：SLA 违反率模型。

SLA 违反率模型定义为一个四元组

$$VioRateSLA = (resp_i, resp_{SLA}, NumVio, TotalVio)$$

SLA 违反率模型描述如下：

1）$resp_i$ 为虚拟机节点 i 响应用户请求的实际请求响应时间。

2）$resp_{SLA}$ 为 *SLA* 中规定的请求平均响应时间。

3）*NumVio*：为请求的违反次数。

当实际请求响应时间 $resp_i$ 低于 $resp_{SLA}$ 的 80%时，即为服务质量满足了用户需求，用户有较好的自身体验，未违反 *SLA*。否则，反之。用公式（17）表示。

$$Vio_{SLA} = \begin{cases} 0 & resp_i \geqslant resp_{SLA} \times 80\% \\ 1 & resp_i < resp_{SLA} \times 80\% \end{cases} \tag{17}$$

根据公式（17），当 Vio_{SLA} 的值为 0 时，用户请求超时，违反了 *SLA*，用户体验效果不好，违反次数增加 1。

4）*TotalVio*：为用户请求的总次数。

基于以上的描述，将请求 *SLA* 违反率函数用公式（18）表示。

$$VioRate_{SLA} = \frac{NumVio}{TotalVio} \tag{18}$$

4 实验结果及分析

4.1 实验环境与算法的参数配置

本文采用 CloudSim 仿真模拟云环境，仿真实验环境参数配置如表 1 所示，同时部署 6 种性能各异的 100 台虚拟机，其参数配置如表 2 所示。

表 1 仿真实验环境参数

实验环境	描述
操作系统	Windows 8.1
虚拟机管理程序	Xen
虚拟机数量	100
应用程序	Web 服务

表 2 虚拟机类型的参数

虚拟机类型	1	2	3	4	5	6
CPU	1	2	2	1	3	2
内存(MB)	512	512	1024	1024	2048	2048
磁盘(GB)	20	20	20	20	20	20
网络带宽(Mb. s^{-1})	5	5	10	15	10	15

SLPA 和 BPNNA－DWRRA 算法参数配置如表 3 和表 4 所示。

4.2 负载状态分类实验结果

SLPA 算法实现负载状态分类，实验结果如图 4 所示。

从图 4 可以看出，直线将坐标轴中的点划分成了两类，直线上方的点为是重载节点，直线下方的点为轻载节点，实验结果表明 SLPA 算法能准确地将虚拟机进行负载状态分类。

表 3 SLPA 算法参数

变量	初始化值
初始权重	ω=[0,0,0]
学习率 η	0.4
ε	0.001
迭代次数 n	1
实际输出函数 f	hardlim

表 4 BPNNA－DWRRA 算法参数

变量	初始化值
误差容限 δ	0.01
学习率 σ	0.6
动量因子 λ	0.8
迭代次数 e	1
传递函数	logsig

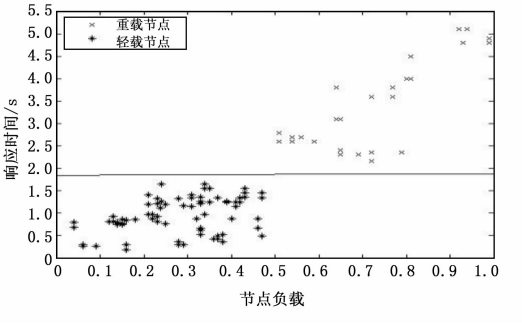

图 4 SLPA 算法负载分类

4.3 负载权重预测更新实验结果

从图 4 即可得到所有轻载节点，采集所有轻载节点的历史权重，利用 BPNNA－DWRRA 算法对这些轻载节点进行权重的预测更新。其实验结果如图 5 所示。

从图 5 可以看出，随着迭代次数的增加，误差逐渐趋于 0，而准确率逐渐趋于 1，表明用权重来指导请求调度的准确率较高，能为用户找到合适的虚拟机资源，证明了该模型的可行性。

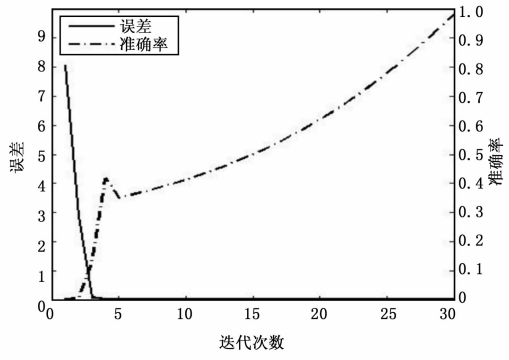

图 5　BPNNA－DWRRA 算法负载权重预测更新

为了说明本文算法具有高效性和 SLA 违反率低，将本文算法与加权最小链接算法[6]和粒子群算法[10]进行比较。其实验结果如图 6 和如图 7 所示。图 6 是这 3 种算法平均响应时间对比图，图 7 是这 3 种算法违反次数对比图。

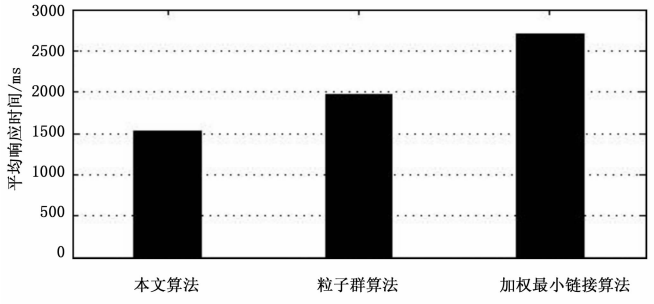

图 6　算法平均响应时间对比图

从图 6 可以看出，本文算法平均响应时间最短，加权最小链接算法算法的平均响应时间总是最长的，粒子群算法在本文算法和加权最小链接算法之间，本文算法的平均响应时间比其他两种算法的平均响应时间分别节省了 43.6％和 22.5％，本文算法能快速的为用户分配请求，具有高效性。

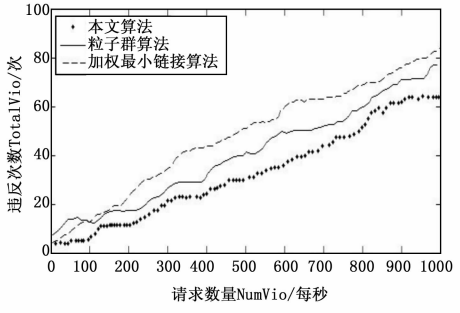

图 7　算法违反次数对比图

从图 7 可以看出，刚开始的时候 3 种算法的违反次数都很接近，而且加权最小链接算法和粒子群算法还有重叠的地方，表明在请求数量少的情况下，3 种算法违反次数都比较少，但是随着请求数量的增加，3 种算法的违反次数出现了明显的差异，加权最小链接算法的违反次数总是最多的，粒子群算法趋于加权最小链接算法和本文算法之间。总之，加权最小链接算法的违反次数最多，粒子群算法违反次数次之，本文算法的违反次数最少。根据违反次数，计算 SLA 违反率，基本上本文算法的违反率比其他两种算法违反率分别降低了 20.7％和 14.4％。即本文算法 SLA 违反率低，保障了 SLA，使用户有较好的使用体验。

5　结论

针对负载均衡问题，考虑了用户在消费云服务时的自身体验，保证其可以接受的服务质量，本文提出了一种面向 SLA 的云计算负载均衡策略。该模型通过 SLPA 算法的虚拟机节点分类，缩小虚拟机节点预测更新范围，提高 BPNNA－DWRRA 算法预测效率，将任务调度到合适的虚拟机上。通过实验结果表明，本文策略实现了负载均衡，同时，与其他几种算法相比，本文算法平均响应时间短，SLA 违反率低，实现了用户的友好体验。

参考文献：

[1] Patel P，Ranabahu A，Sheth A. Service Level Agreement in Cloud Computing [J]. Cloud Sla，2009.

[2] Lin C C，Chin H H，Deng D J. Dynamic Multiservice Load Balancing in Cloud－Based Multimedia System [J]. Systems Journal IEEE，2014，8 (1)：225－234.

[3] 田绍亮，左　明，吴绍伟. 一种改进的基于动态反馈的负载均衡算法 [J]. 计算机工程与设计，2007，28 (3)：572－573.

[4] 孙峻文，周　良，丁秋林. 基于退火算法的动态负载均衡研究 [J]. 计算机科学，2013，40 (5)：89－92.

[5] 宁　彬，谷　琼，吴　钊，等. 云计算环境下的混沌萤火虫的资源负载均衡算法 [J]. 计算机应用研究，2014 (11).

[6] 蔡程宇，娄渊胜. 改进加权最小连接数负载均衡调度算法研究 [J]. 哈尔滨商业大学学报（自然科学版），2015 (1)：102－104.

[7] Xiong，Yeliang，Cang. Task scheduling and virtual machine allocation policy in cloud computing environment [J]. Journal of Systems Engineering & Electronics，2015 (04)：847－856.

[8] Cho K M，Tsai P W，Tsai C W，et al. A hybrid meta－heuristic algorithm for VM scheduling with load balancing in cloud computing [J]. Neural Computing & Applications，2015，26 (6)：1297－1309.

[9] 王常芳，徐文忠. 一种用于云计算资源调度的双向蚁群优化算法 [J]. 计算机测量与控制，2015，23 (8)：2861－2863.

[10] Ramezani F，Lu J，Hussain F K. Task－Based System Load Balancing in Cloud Computing Using Particle Swarm Optimization [J]. International Journal of Parallel Programming，2014，42 (5)：739－754.

[11] Ye F，Wang Z J，Xu F，et al. A Novel Cloud Load Balancing Mechanism in Premise of Ensuring QoS [J]. Intelligent Automation & Soft Computing，2013，19 (2)：151－163.

[12] 刘智斌，曾晓勤，刘惠义等. 基于 BP 神经网络的双层启发式强化学习方法 [J]. 计算机研究与发展，2015，(3)：579－587.

[13] 刘建伟，申芳林，罗雄麟. 感知器学习算法研究 [J]. 计算机工程，2010，36 (7)：190－192.

[14] 江　丽，王爱平. 基于粒子群与 BP 混合算法的神经网络学习方法 [J]. 计算机应用，2012，32 (S2)：13－15.