基于负载预测的自适应权值负载均衡算法

王宇耕[1]，肖 鹏[1]，张 力[1]，王激扬[2]，孙 涛[3]

（1. 中国航天科工集团第二研究院 706 所，北京 100854；2. 海军装备部驻北京地区军事代表局，北京 100071；3. 中国长峰机电技术研究设计院，北京 100854）

摘 要：针对 LVS 集群的加权最小连接数算法在反映实际负载和具体性能方面的缺陷，提出一种基于负载预测的自适应权值负载均衡算法。采用一种自适应 AR 算法计算负载预测值，使服务器权值能够根据预测结果做出调整，进一步提升负载均衡效果，更加有效地利用系统资源。实验采用 WAS 工具进行仿真，并对算法效果进行对比，实验结果表明，该算法能够更好地在 LVS 集群系统中实现负载均衡。

关键词：负载均衡算法；负载预测；自适应 AR；LVS 集群；自适应权值

中图法分类号：TP391 **文献标识号**：A **文章编号**：1000-7024（2019）04-1033-05

doi：10.16208/j.issn1000-7024.2019.04.022

Adaptive weight load balancing algorithm based on load prediction

WANG Yu-geng[1], XIAO Peng[1], ZHANG Li[1], WANG Ji-yang[2], SUN Tao[3]

(1. Institute 706, Second Academy of China Aerospace Science and Industry Corporation, Beijing 100854, China;
2. Military Representative Bureau of Naval Equipment Department in Beijing, Beijing 100071, China;
3. Academy of China ChangFeng Electromechanical Technology, Beijing 100854, China)

Abstract: Aiming at the problem that the weighted least connection algorithm can not accurately reflect the actual load and processing capacity of the server, a load balancing algorithm based on load prediction and adaptive weight was proposed. An adaptive AR algorithm was used to calculate the load prediction value, so that the server weights were adjusted according to the prediction results, further improving the load balancing effect and making the most of the system resources. The experiment was simulated with WAS tool, and the effects of the algorithm were compared. Experimental results show that the proposed algorithm can achieve better load balancing effect in the LVS cluster system.

Key words: load balancing algorithm; load prediction; adaptive AR; LVS cluster; adaptive weight value

0 引 言

负载均衡算法是负载均衡集群的核心，目前 LVS 集群在内核模块已经集成了 10 种不同的调度算法[1]。在部署 LVS 集群时，通常采用动态负载均衡算法作为任务分配的策略，其中负载均衡效果最好的是加权最小连接数算法[2]。加权最小连接数算法以权值评定服务器处理能力，以任务连接数衡量服务器负载，在一定程度上较好地实现了负载均衡效果。但是加权最小连接数算法有两种缺陷：①权值为固定值，而服务器的处理性能会因负载变化而变化。②算法以任务连接数为负载状态依据，而实际的负载状态不能仅用连接数表示。当前国内外许多研究人员针对上述缺陷做出了一定的改进，如 Sun D 等[3]针对第二种缺陷，以历史负载信息作为负载因子来影响任务分配，这种算法以实际负载作为依据，与原始算法相比拥有更好的负载均衡效果；Wu Y 等[4]提出一种自适应权值的动态负载均衡算法，该算法考虑了权值变化对服务器的影响，可以有效避免负载过量问题。当前的改进算法通常是以固定周期采集负载信息或者根据一段时间内的负载状态作为任务分配的依据，而负载是时刻在变化的，如果负载信息采集的时间周期设置不合理就有可能会有时延问题，从而影响了算法的实际应用。如果能够掌握服务器负载未来变化趋势，以

收稿日期：2017-12-27；**修订日期**：2018-03-05

作者简介：王宇耕（1990-），男，北京人，硕士研究生，研究方向为集群并行计算、负载均衡；肖鹏（1968-），男，北京人，硕士，研究员，研究方向为嵌入式计算机、集群系统等；张力（1986-），男，北京人，硕士，高级工程师，研究方向为嵌入式系统等；王激扬（1968-），男，江苏常州人，博士，高级工程师，研究方向为系统工程与数据挖掘；孙涛（1980-），男，辽宁沈阳人，硕士，工程师，研究方向为嵌入式系统与智能信息处理。E-mail：wangyugeng2006@sina.com

负载信息的预测值作为任务分配的依据，就可以减少时延问题的影响。因此，本文在原算法基础上加入负载预测环节，提出了基于负载预测的自适应权值负载均衡算法，使提出算法中的权值能够根据预测结果动态调整。

1 负载预测

1.1 服务器负载计算

负载是服务器当前处理能力的体现，如果能够获知负载在将来的变化，就相当于获知了服务器的处理性能在将来的变化。如果以某一固定周期对服务器进行采样，那么采样数据可以看作一组负载的时间序列。在异构集群中，为了使负载均衡器能更均衡地分配任务，充分利用集群系统的资源，必须准确地获知服务器节点当前的负载[5]，为此需要一个函数将异构集群中不同服务器的各项负载信息转化为综合负载状态信息[6]。本文以服务器的综合负载指标 $L(i)$ 作为负载状态的识别依据[7]，选取 4 项指标（CPU 利用率、内存使用率、磁盘 I/O 占用率和网络带宽使用率）作为评价负载状态的指标，转化函数表达式如下所示

$$L(i)=\begin{bmatrix}r_1 & r_2 & r_3 & r_4\end{bmatrix}\begin{bmatrix}L_C(i) & L_M(i) & L_D(i) & L_N(i)\end{bmatrix}^T$$

式中：$L_C(i)$、$L_M(i)$、$L_D(i)$、$L_N(i)$ 代表上述 4 项指标的负载数值，参数 r 代表各部分负载对 $L(i)$ 的影响程度，且 $\sum r_i=1$，影响程度更大的指标的 r 值更高[8]。

1.2 自适应 AR 模型

利用采集到的服务器历史负载数据，本文采用一种自适应 AR 算法对负载预测值进行计算[9]。服务器负载预测的 AR(P) 模型可以表示为

$$l_t=\sum_{i=1}^{p}\varphi_i l_{t-i}+\varepsilon_t,\ t\in Z \tag{1}$$

式中：l_t 代表 t 时刻的负载预测值，l_t 可由 t 时刻前的 p 个实际负载值计算得出，p 为模型阶数，$\varphi_i(i=1,2,3,\cdots,p)$ 是 AR 模型的自回归系数，可以通过某一种参数估计的方式得出[10]，l_{t-i} 是 t 时刻之前的实际负载数据，ε_t 为残差。如果动态地调整自回归系数 φ_i，使 ε_t 的方差最小，则上式相当于最小方差拟合的 AR(p) 模型。对于服务器负载预测来说，自回归系数 φ_i 是一个不确定的值，φ_i 会根据负载数据的更新不断调整。式（1）中，自回归系数的向量形式表示为

$$\Phi=\begin{bmatrix}\varphi_{1t} & \varphi_{2t} & \cdots & \varphi_{pt}\end{bmatrix} \tag{2}$$

负载数据的向量形式表示为

$$L_t=\begin{bmatrix}l_{t-1} & l_{t-2} & \cdots & l_{t-p}\end{bmatrix} \tag{3}$$

那么，负载预测的自适应 AR 模型的矩阵表达式为

$$l_t=\Phi_t L_t{}^T+\varepsilon_\tau \tag{4}$$

如果能够为阶数 P 和模型的自回归系数 Φ 选取一个合理的数值，就能够较为精确的计算出负载的预测值[11]。

1.2.1 参数初始值的估计

自适应 AR(P) 模型的参数估计中，阶数 p 的选取十分重要。阶数直接影响了 AR 模型预测结果的精度和预测时间。阶数过低会导致误差过大，不能满足负载均衡系统的预测需求。阶数过高不仅会增加预测时间，浪费系统资源，过高的阶数也会导致虚假峰值产生。只有为 p 选择一个合理的数值，才能将预测效果发挥最佳。合理的 p 值应该是使预测误差期望最小的阶数。当前模型定阶有多种判别准则，常用的准则有 SC 准则、FPE 准则和 AIC 准则等，因为 p 是一个整数值，由不同的准则得到的最佳 p 值可能有所不同，本文以 AIC 准则估计最佳阶数 p 的取值，AIC 准则表达式如下

$$\mathrm{AIC}(p)=\ln\hat{\sigma}^2+2p/N \tag{5}$$

AIC 准则函数由两项组成[12]，第一项为残差方差的函数，$\hat{\sigma}^2$ 的值随着 p 的增加会减少，因此 $\ln\hat{\sigma}^2$ 会随着 p 的增加单调下降，而第二项则随着 p 的增加而单调增加，因此可以确定某一阶数 p，若 p 能够使 AIC 函数取得最小值，则此阶数 p 为最佳阶数。

确定模型阶数 p 之后，通过 Yule-Walker 方程对自回归系数进行计算。Yule-Walker 的数学表达式如下

$$\begin{bmatrix}1 & \rho_1 & \cdots & \rho_{p-1}\\ \rho_1 & 1 & \cdots & \rho_{p-2}\\ \vdots & \vdots & \vdots & \\ \rho_{p-1} & \rho_{p-2} & \cdots & 1\end{bmatrix}\begin{bmatrix}\Phi_1\\ \Phi_2\\ \vdots\\ \Phi_p\end{bmatrix}=\begin{bmatrix}\rho_1\\ \rho_2\\ \vdots\\ \rho_p\end{bmatrix} \tag{6}$$

式中：ρ_k 为样本的自相关系数。自相关系数可由下式所得

$$\rho_k=\frac{\sum_{t=1}^{n-k}(x_t-\bar{x})(x_{t+k}-\bar{x})}{\sum_{t-1}^{n}(x_t-\bar{x})^2},\ 0<k<n \tag{7}$$

通过式（5）～式（7）就可以完成负载预测的 AR(P) 模型的初始建模。

1.2.2 参数 Φ 的自适应调整

负载预测既应考虑到预测精度，也应考虑到实时性要求。由于服务器负载会受到各方面因素的影响是一个时刻变化的数值，如果仅采用预先设定好的 AR 模型不能很好适应该变化，所以模型参数需能够根据实际的负载情况自适应调整。综合考虑预测精度、计算成本和实时性，本文采用一种变步长的 LMS 算法对自回归系数自适应估计。

由式（4）可得残差的平方和残差的均方值

$$\varepsilon_t{}^2=l_t{}^2-2l_tL_t{}^T\Phi_t+\Phi_t{}^TL_tL_t^T\Phi_t \tag{8}$$

$$E(\varepsilon_t{}^2)=E(l_t{}^2)-2R_{lL}{}^T\Phi_t+\Phi_t{}^TR_{LL}\Phi_t \tag{9}$$

其中

$$R_{lL}\triangleq E[l_tL_t]=E\begin{bmatrix}l_{t-1}l_t\\ l_{t-2}l_t\\ \vdots\\ l_{t-p}l_t\end{bmatrix} \tag{10}$$

$$R_{LL}\triangleq E[L_tL_t^T]=E\begin{bmatrix}l_{t-1}l_{t-1} & l_{t-1}l_{t-2} & \cdots & l_{t-1}l_{t-p}\\ l_{t-2}l_{t-1} & l_{t-2}l_{t-2} & \cdots & l_{t-2}l_{t-p}\\ \vdots & \vdots & & \\ l_{t-p}l_{t-1} & l_{t-p}l_{t-2} & \cdots & l_{t-p}l_{t-p}\end{bmatrix} \tag{11}$$

由式（9）可知，预测负载的均方误差是自回归系数 Φ 的二次函数。根据梯度法，使均方误差最小的最佳自回归系数向量 Φ_{LMS} 可表示为

$$\Phi_{LMS} = {R_{LL}}^{-1} R_{lL} \tag{12}$$

由于直接求解 Φ_{LMS} 比较困难，本文采用最速下降法求解。根据最速下降法的思想，$t+1$ 时刻的自回归系数向量 Φ_{t+1} 等于 t 时刻的自回归系数向量 Φ_t 与一个均方误差梯度的比例项的相减值

$$\Phi_{t+1} = \Phi_t - \mu \nabla(t) \tag{13}$$

在实际计算中采用估计值式 $\hat{\nabla}(t)$ 代替 $\nabla(t)$

$$\hat{\nabla}(t) = \hat{\nabla} E({\varepsilon_t}^2) \approx \nabla[{\varepsilon_t}^2] = 2\varepsilon_t \nabla[\varepsilon_t] \tag{14}$$

由式（14）和式（4）可推导出自回归系数的自适应递推公式为

$$\Phi_{t+1} = \Phi_t + 2\mu\varepsilon_t L_t \tag{15}$$

式中：μ 是步长因子，$\mu \in (0, 1/\lambda_{max})$，$\lambda_{max}$ 是历史负载数据的自相关矩阵的最大特征值。在 LMS 算法中，步长因子与收敛速度和稳态误差密切联系，为权衡收敛速度和稳态误差，本文采用了一种变步长的方法[13]，步长因子 μ 可表示为

$$\mu = \lambda_{\max} - \frac{1}{\sqrt{c\left\|\varepsilon_t L_t\right\| + (1/\lambda_{\max})^2}} \tag{16}$$

式中：c 为一常量。因此，通过式（4）、式（15）、式（16），可以对负载预测模型的自回归系数进行自适应调整，从而完成负载的预测。

1.3 负载预测实现步骤

根据上述分析，自适应 AR 的负载预测步骤为：

（1）服务器进入工作状态后，以某一固定周期进行采样并转化为综合负载信息，生成负载数据的时间序列。

（2）对负载数据的时间序列采用 AIC 准则确定模型的阶数 p，然后由 Yule-Walker 法确定模型的自回归系数的初始值 Φ_0。

（3）利用变步长的 LMS 算法，根据 t 时刻前 p 个负载数据，由式（15）、式（16）确定 t 时刻的自回归系数 Φ_t，带入到式（4）中，得到 t 时刻的负载预测值 l_t；

（4）根据服务器 t 时刻真实负载，更新负载时间序列中的数据，为下一次计算做准备。

（5）$t+1$ 时刻，重复（3）、（4）计算步骤。

2 基于负载预测的自适应权值负载均衡算法

2.1 算法基本思想

根据第一节的分析，利用自适应 AR 算法可以较为准确地得到各个节点的负载预测值。为了更好实现负载均衡效果，本文结合负载预测和加权最小连接数算法，提出一种根据负载预测值进行权值自适应调整的改进算法。本文所提算法以负载的预测值作为依据，动态地对权值进行调整，使任务分配获得提前量。负载均衡器按照自适应调整后的新权值对任务进行分配，对未来负载变重的服务器的初始权值进行下调，对未来负载变轻的服务器初始权值进行上调，从而可以在一定程度上避免时延问题。

2.2 算法实现

2.2.1 服务器初始权值计算

为实现本文所提出的调度算法，首先对服务器初始权值进行设置。服务器权值设置应考虑不同部分的性能对服务器整体性能的影响，这些性能信息需要通过函数进行转化，初始权值的表达式如下所示

$$W(i) = [m_1 \quad m_2 \quad m_3 \quad m_4] \quad [M_C(i) \quad M_D(i) \quad M_M(i) \quad M_N(i)]^T \tag{17}$$

式中：$\sum m_i = 1$，$W(i)$ 为初始权值，$M_C(i)$、$M_D(i)$、$M_M(i)$、$M_N(i)$ 分别代表 CPU 性能、内存容量、磁盘 I/O 速率和网络吞吐能力，参数 m 用以强调各部分对于服务器整体性能的影响力。在 LVS 集群部署中，初始权值的设置往往依赖于服务器管理员的经验，由于在最小加权连接数算法中权值是不变的，如果权值设置的不够合理，则直接影响负载均衡效果[14]。

2.2.2 权值的动态调整

权值的调整应以集群系统稳定运行为前提，在实际工作中，服务器的处理能力在短时间内不会有太大波动，因此不需要频繁调整服务器权值。如果服务器的预测负载值处于相对正常的状态，服务器当前权值就无需调整，只有当预测负载值处于过高或者过低的状态，才需要对权值进行调整[15]。

为此，本文以一种分段计算的动态权值公式来计算新权值。我们首先设定综合负载指标的 3 个取值 L_l、L_h、L_{max} 分别作为低负载、高负载和饱和负载的数值，定义阈值范围 $(0, L_l)$ 代表服务器的低负载区，(L_l, L_h) 代表服务器的正常负载区，(L_h, L_{max}) 代表服务器的高负载区，$(L_{max}, 1)$ 代表服务器的饱和负载区。通过第一节中所提的负载预测算法，我们可以得到 t 时刻的负载预测值 l_t，在计算新的权值的时候，若 $l_t \in (0, L_l)$，可认为服务器将在 t 时刻处于空闲状态，应该适当地加大权值，发挥服务器的处理性能；若 $l_t \in (L_l, L_h)$ 时，可以认为服务器在 t 时刻处于正常负载状态，可以继续接纳新任务，权值无需调整；当 $l_t \in (L_h, L_{max})$ 时，说明服务器在 t 时刻处于高负载状态，需要适当减小服务器的权值，避免出现负载过重的情况，而如果 $l_t \in (L_{max}, 1)$ 时，说明服务器的负载已经接近饱和了，此时应避免接纳新任务，不管 l_t 超出 L_{max} 多少，都把权值设置为 0，直到之后的某个时刻 $t+k$ 的预测值 L_{t+k} 小于 L_{max}，重新计算权值。由上所述，新权值的计算公式如下

$$W_n(i) = \begin{cases} W(i) + M * (L_l - l_t(i)) & 0 < l_t < L_l \\ W(i) & L_l < l_t < L_h \\ W(i) - N * (L_{max} - l_t(i)) & L_h < l_t < L_{max} \\ 0 & L_{max} < l_t < 1 \end{cases} \tag{18}$$

式中：$W(i)$ 为初始权值，由式（18）计算得到。M 和 N 为两个可调节参数，本文设置为 10 和 8，并规定 L_l、L_h、L_{max} 分别为 0.6、0.8、0.9。举例分析，假设当前存在两台服务器 S_1，S_2，初始权值比为 8∶10，预测负载值分别为 0.3 和 0.7，意味着 S_1 将处于较为空闲的状态，需要调高权值。则通过计算公式得到的新权值 $W_n(1)=11$，$W_n(2)=10$，此时权值比由 8∶10 变为 11∶10，因此在新任务到来时，服务器 S_2 相比服务器 S_1 将获得更高的优先级。这种动态自适应调整的权值与静态权值相比，能够使得负载均衡更为有效。

为了更加准确地对负载均衡进行控制，动态权值的调整也应当考虑到服务器的某项具体负载水平，本文对权值调整加入约束条件：当 4 项负载预测指标（CPU 使用率、内存使用率、磁盘 I/O 使用率、带宽使用率）任意一项处于 $(L_{max},1)$ 区间内，$W_n(i)=0$。

2.2.3 任务分配过程

假设当前集群系统内有 n 台性能不同的服务器，S_i 代表第 i 台服务器，初始权值为 $W(i)$，当前时刻为 t，已知各台服务器的任务连接数为 $C(i)$。当有新任务在 $(t,t+1)$ 时刻到达时，任务分配过程如下：

（1）根据 t 时刻之前的负载数据的时间序列，由自适应 AR 算法得出 t 时刻的负载预测值 l_{t+1}。负载均衡器收到预测结果 l_{t+1} 后，判断是否有必要调整权值，为预测负载值过高或者过低的服务器计算新权值，将预测负载值处于饱和状态的服务器的权值置为 0，更新服务器列表。

（2）负载均衡器根据更新后的服务器列表对服务器 S_i 进行轮询，获取当前服务器的连接数和新权值。选择满足条件：$[(C(m)/C_{sum})/W_n(m)]=min[(C(i)/C_{sum})/W_n(i)]$ 的服务器 m 作为负载最轻的服务器来分配任务。

（3）服务器处理完任务之后，在 $t+1$ 时刻，根据自身负载值更新预测模型参数，重复上述计算步骤，重新计算权值并完成任务分配。

3 算法性能分析

为验证算法性能，本文搭建了 LVS 集群作为实验环境。LVS 集群由 3 台搭载 ubuntu14.04 系统的计算机组成，其中一台作为 LVS 集群的负载均衡器，另外两台作为真实服务器提供相同的服务，初始权值比为 8∶10。另选一台搭载 windows7 系统的计算机作为客户端进行测试。压力测试工具为 Microsoft Web Application Stress Tool（WAS）。

实验通过模拟多个客户端同时对服务器集群进行访问，形成高并发环境对服务器产生负载压力。实验通过对两个方面的性能来衡量算法效果：①每秒处理的请求数：每秒处理的请求数越多，这个数值越大，说明集群系统的在单位时间内的处理任务的数量越多，吞吐率越高。②响应时间：WAS 测试收到真实服务器回复的第一个数据所用的时间，响应时间越小，说明集群的性能更好。

将 WAS 并发测试时间设置为 5 分钟，用户访问连接数从 100 到 600 分为 6 组进行实验，分别对加权最小连接数算法和本文提出的基于预测的动态负载均衡算法进行仿真实验。通过 WAS 测试工具，可以分别得出客户端单位时间收到的字节数和响应时间。实验测试数据如下，单位时间收到的字节数见表 1，响应时间见表 2。

表 1 单位时间收到的字节数/(KB/s)

连接数	加权最小连接数算法	本文算法
100	2.3	2.1
200	3.9	3.7
300	5.7	5.6
400	8.9	9.1
500	11.1	11.7
600	12.7	13.5

表 2 响应时间/ms

连接数	加权最小连接数算法	本文算法
100	30.7	36.2
200	57.6	61.2
300	240.1	228.1
400	612.1	564.7
500	1346.7	1017.2
600	2278.5	1679.1

由表 1 及表 2 的数据，可以得到实验结论：当并发连接数在 100 到 400 时，即用户连接数量相对较少时，本文所提算法与系统原有算法相比，在单位时间收到的字节数和响应时间上相差不大，并且在 100 到 300 低负载段，由于本文所提算法需要额外消耗服务器资源进行负载预测值的计算，在一定程度上会对服务器增加少量负担，所以当连接数比较少时，所提算法不如原有算法。但是，并发连接数在 500 到 600 时，即集群系统相对处于较高负载时，负载预测带来的少量资源消耗相比服务器整体的负载可以忽略不计，从实验结果可以看出此时本文所提算法好于原有算法，具有更好的负载均衡效果。通过上面的分析，我们可以得出本文所提出的基于负载预测的自适应权值动态负载均衡算法，在并发连接数比较大的情况下要优于原有的算法。

4 结束语

本文针对加权最小连接数算法在反映实际负载和处理能力上的缺陷，提出了一种基于负载预测的自适应权值负载均衡算法。该算法是加权最小连接数算法的改进，在原

有算法基础上加入预测环节，采用自适应 AR 算法，利用历史负载数据，对负载预测值进行估计，根据负载预测的结果，采用一种权值计算方式重新计算权值，新权值能够根据负载变化提前做出调整，使负载均衡器最大化发挥负载均衡作用。实验结果表明，本文所提算法与原始算法相比，在并发数相对较多时，在负载均衡效果上有一定的提升。

参考文献：

[1] Zhang Y，Han Y，Sun G. Research on the construction of high available load balancing cluster based on LVS [C] // Workshop on Advanced Research & Technology in Industry Applications，2016.

[2] WANG Chunjuan. Research and analysis of LVS cluster algorithm [J]. Computer Knowledge and Technology，2013，9 (26)：5963-5964 (in Chinese). [王春娟. LVS 集群算法研究与分析 [J]. 电脑知识与技术，2013，9 (26)：5963-5964.]

[3] Sun D，Zhu Q. Comparison and improvement of load balance scheduling algorithm based on cluster technology [C] //International Conference on Electrical and Electronics Engineering，2014：67-74.

[4] Wu Y，Luo S，Li Q. An adaptive weighted least-load balancing algorithm based on server cluster [C] //International Conference on Intelligent Human-Machine Systems and Cybernetics，2013：224-227.

[5] Kanakala R，Reddy VK. Performance analysis of load balancing techniques in cloud computing environment [J]. Telkomnika Indonesian Journal of Electrical Engineering，2015，9 (18)：1-6.

[6] Ren X，Lin R，Zou H. A dynamic load balancing strategy for cloud computing platform based on exponential smoothing forecast [C] //IEEE International Conference on Cloud Computing and Intelligence Systems. IEEE，2011：220-224.

[7] Hu Y，Zhu S. Load-balancing cluster based on Linux Virtual Server for internet-based laboratory [C] //Industrial Electronics and Applications. IEEE，2014：2181-2185.

[8] Bosque JL，Toharia P，Robles OD，et al. A load index and load balancing algorithm for heterogeneous clusters [J]. Journal of Supercomputing，2013，65 (3)：1104-1113.

[9] Liu C，Hoi SCH，Zhao P，et al. Online ARIMA algorithms for time series prediction [C] //30th AAAI Conference on Artificial Intelligence. AAAI Press，2016：1867-1873.

[10] Hua ZQ，Xue DM. ARIMA based time series forecasting model [J]. Recent Advances in Electrical & Electronic Engineering，2016，9 (2)：93-98.

[11] ZHANG Zonghua，ZHANG Haiquan，WEI Chi，et al. LVS cluster load balancing algorithm with adaptive weight leastload [J]. Communications Technology，2016，24 (3)：248-251 (in Chinese). [张宗华，张海全，魏驰，等. 基于加权改进的 AR 模型的负载预测研究 [J]. 计算机测量与控制，2016，24 (3)：248-251.]

[12] ZHAO Ming. Research on network bandwidth allocation algorithm based on AR(n) model [J]. Microcomputer Applications，2017，33 (6)：61-63 (in Chinese). [赵明. 基于 AR(n)模型的网络带宽流量动态分配预测算法研究 [J]. 微型电脑应用，2017，33 (6)：61-63.]

[13] HUAN Qiuyun，QIU Xiaohui，LIU Xiaofei. Variable step LMS algorithm using norm of the hyperbolic tangent function [J]. Journal of Signal Processing，2014，30 (1)：93-99 (in Chinese). [还秋云，邱晓晖，刘晓飞. 引用范数的双曲正切函数变步长 LMS 算法 [J]. 信号处理，2014，30 (1)：93-99.]

[14] QIN Xia. The design and implementation on LVS load balancing cluster [D]. Xi'an：Xidian University，2014 (in Chinese). [秦霞. 基于 LVS 负载均衡集群的设计和实现 [D]. 西安：西安电子科技大学，2014.]

[15] YANG Ting，WAN Liang，MA Shaoju，et al. LVS cluster load balancing algorithm with adaptive weight leastload [J]. Communications Technology，2017，50 (4)：741-745 (in Chinese). [杨婷，万良，马绍菊，等. 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，2017，50 (4)：741-745.]