一种改进的 Kubernetes 弹性伸缩策略*

沐　磊　李洪赭　李赛飞
（西南交通大学信息科学与技术学院　成都　611756）

摘　要　Kubernetes是目前主流的容器云编排和管理系统，其内置的伸缩策略是通过监测衡量指标并与阈值比较计算，从而实现伸缩的功能。该策略主要存在单一衡量指标和响应延迟问题：单一指标在衡量多种资源消耗的复杂应用时存在明显缺陷；响应延迟问题会造成应用在一段时间内的服务质量无法得到保障。针对上述问题，提出了一种改进的Kubernetes弹性伸缩策略，该策略对应用涉及的多种资源进行计算，得出综合负载率作为衡量应用伸缩的指标；使用ARIMA-Kalman预测模型对综合负载进行预测，从而实现预测式伸缩，提高了应用在面对突发流量的应对能力。实验结果表明，该策略能够较好地衡量应用的整体负载水平，并能对应用负载进行准确的预测，解决了响应延迟的问题。

关键词　Kubernetes；弹性伸缩；综合负载；ARIMA模型；卡尔曼滤波
中图分类号　TP319　　**DOI**：10. 3969/j. issn. 1672-9722. 2022. 02. 020

An Improved Kubernetes Elastic Scaling Strategy

MU Lei　LI Hongzhe　LI Saifei
（School of Information Science and Technology，Southwest Jiaotong University，Chengdu　611756）

Abstract　Kubernetes is currently the mainstream container cloud orchestration and management system. Its built-in scaling strategy is to achieve scaling by monitoring metrics and comparing with thresholds. The strategy mainly has a single measurement index and response delay，a single indicator has obvious shortcomings when measuring complex applications that consume multiple resources. The response delay will cause the application's service quality to be unable to be guaranteed for a period of time. In view of the above problems，an improved Kubernetes elastic scaling strategy is proposed. This strategy calculates multiple resources involved in the application and obtains the comprehensive load rate as an index to measure the application scalability. The ARIMA-Kalman prediction model is used to predict the comprehensive load，so as to achieve predictive scaling and improve the application's ability to cope with sudden traffic. Experimental results show that the strategy can well measure the overall load level of the application，and can accurately predict the application load，and the problem of response delay is solved.

Key Words　Kubernetes，elastic scaling，comprehensive load，ARIMA model，Kalman filtering method
Class Number　TP319

1　引言

随着虚拟化技术和云计算技术的快速发展，基于Hypervisor传统虚拟化的云计算技术因为其较低的资源利用率等一系列问题，已经逐渐被以Docker[1]为代表的基于容器虚拟化的容器技术所代替[2]。相比于传统虚拟化技术，Docker容器技术通过复用本地操作系统来提高启动速度，减少开销。同时因为Docker容器技术简化了应用的部署工作，受到广大开发者的欢迎[3]。然而在面对大规模集群的容器组时，对这些容器进行管理会显得非常困难。Kubernetes[4]是Google的Borg[5]开源版本，Docker容器集群编排调度系统，使用Master-Slave模型[6]为容器化应用提供资源调度、自动化部署、服务发现、弹性伸缩、资源监控等服务[7]。其中弹性伸缩是通过监测用户指定的评价指标，使用阈值的方式对应用进行水平扩容和缩容，以保障应用的服务质量和最大化地节省资源。

* **收稿日期**：2021年7月9日，**修回日期**：2021年8月20日
基金项目：四川省重大科技专项课题（编号：2018GZDX0005，2019YFG0399，2019ZDZX0007）；中央高校基本科研业务费专项（编号：2682019CX63）资助。
作者简介：沐磊，男，硕士研究生，研究方向：云计算与网络。李洪赭，男，博士研究生，研究方向：网络与信息安全。李赛飞，男，博士研究生，工程师，研究方向：铁路信号系统网络安全。

针对Kubernetes的弹性伸缩技术的研究大致可以分为两类,第一类采用方法或者预测模型对未来资源的利用率进行估计或预测。文献[8]针对扩容和缩容时的不同特点,分别提出了快速扩容算法和逐步收缩算法,通过在工作负载增加时快速创建多个pod和在工作负载降低时逐步缩容的方式保障应用的服务质量。文献[9]使用了基于动态参数的指数平滑法对资源利用率进行预测,通过预测式扩容和响应式扩容相结合的办法解决Kubernetes扩容时的响应延迟问题。这些处理方法虽然简单,但是在提高资源利用率和预测精度上仍有不足。第二类是针对Kubernetes内置的弹性伸缩策略进行改进和优化。文献[10]提出了一种基于负载特征的弹性伸缩策略,该策略根据负载特征对应用进行分类,求出应用的整体负载并使用ARIMA模型对负载进行预测,使用预测值和负载值共同作为判断伸缩的条件,以减少扩缩容时的伸缩抖动现象。该策略提出的根据负载特征衡量复合型应用负载的方法未考虑各种资源的重要性差异和资源极度不均衡情况下的服务质量问题。

综上所述,虽然目前针对Kubernetes弹性伸缩策略中存在的问题的研究已经比较全面,但是仍有改善的空间。本文通过对Kubernetes内置的伸缩策略进行分析,针对单一衡量指标和预测问题提出了一种改进的弹性伸缩策略。该策略给出综合负载率CLR(Comprehensive Load Rate)的计算方法,使用CLR作为弹性伸缩的指标。同时使用ARIMA-Kalman预测模型对资源进行预测,提高预测精度。

2 Kubernetes默认伸缩算法及分析

Pod是Kubernetes资源控制的基本单位。Kubernetes通过HPA(Horizontal Pod Autoscaler)实现Pod的自动伸缩[11]。HPA控制Kubernetes伸缩原理图如图1所示。

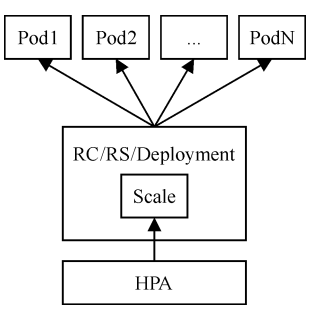

图1 Kubernetes扩缩容原理图

在部署应用时会设置某项资源监测指标和该资源的目标平均使用率阈值 TAU 。则该指标的伸缩阈值计算公式为

$$\begin{cases} up = TAU*(1+tolarence) \\ down = TAU*(1-tolarence) \end{cases} \tag{1}$$

其中 $tolarence$ 系统默认值为0.1,设置该参数是为了防止应用出现频繁地扩容和缩容。 up 和 $down$ 分别是伸缩的上下限阈值。假设当前共有 k 个该应用的pod,HPA通过轮询的方式获取当前该pod集合中所有的该资源使用量 U_i ,求得当前资源利用率和 CAU 为

$$CAU = \sum_{i=1}^{k} U_i / request \tag{2}$$

其中 $request$ 代表该pod中资源的分配量。如果 $k*down \le CAU \le k*up$,则不需要扩缩容,否则需要进行扩缩容操作,计算公式为

$$TPN = ceil(CAU/TAU) \tag{3}$$

其中 TPN 代表目标pod个数, $ceil$ 表示对计算结果向上取整。为了限制pod数量,有如下约束:

$$\begin{cases} TPN < R_{\min} & TPN = R_{\min} \\ TPN > R_{\max} & TPN = R_{\max} \end{cases} \tag{4}$$

以上是Kubernetes实现伸缩功能的流程概述。从上面的分析可以看出。虽然Kubernetes内置的水平伸缩的算法比较简单,但是存在两个比较明显的问题:单一衡量指标和响应延时。在面对复杂应用系统时,应用涉及多种资源类型的消耗可能会随着时间和业务的变化而发生变化,单一衡量指标无法准确地衡量应用整体负载的情况。当应用面对突发的负载变化时,在pod扩展之前的应用服务质量无法保障,甚至可能会发生应用因为负载太高而崩溃的现象。

3 本文伸缩策略优化

针对上文所说的单一衡量指标和响应延时问题,本文提出了一种改进的弹性伸缩策略。该策略通过监测记录pod各个指标,计算出当前pod的综合负载率CLR(Comprehensive load rate),将CLR作为pod伸缩的衡量指标。同时使用ARIMA-Kalman模型对CLR进行预测,提前进行伸缩以保障服务质量。

3.1 衡量指标确定

影响应用服务质量的因素有许多,包括CPU、内存、网络、磁盘I/O等多个基础指标。假设当前pod节点涉及的资源种类为 n , C_i 表示节点上第i种资源的使用率 C_i 为

$$C_i = U_i / R_i \tag{5}$$

其中 R_i 表示节点上资源 i 的分配量，U_i 表示 pod 对资源 i 使用量。记 $C_{\max}$ 为各种资源利用率的最大值，为了减少计算的复杂度，当某项资源利用率低于下限阈值时，认为该资源的利用率对应用负载几乎没有影响，则该资源的利用率将不作为计算综合负载 CLR 的参数。如果 $C_{\max}$ 低于下限阈值，则令 $CLR = C_{\max}$。如果 $C_{\max}$ 高于上限阈值，则资源可以被认为是应用的性能瓶颈，为了保证服务质量，防止出现只有一种资源消耗非常高，其他资源利用率相对较低导致综合负载率没有达到扩容阈值的情况，同样令 $CLR = C_{\max}$。由经验，设定下限阈值 dl =0.4，上限阈值 ul =0.95。

假设当前应用涉及的 n 种资源利用率中有 $k(k>0)$ 个资源利用率满足 $dl < C_i < ul$ 且 $C_{\max} < ul$，则 CLR 为

$$CLR = \sum_{i=1}^{k} m_i C_i \tag{6}$$

否则，$CLR = C_{\max}$。其中 m_i 代表的是第 i 种资源利用率的动态权值。某项负载的利用率越高，对 CLR 的影响越大，所以其动态权值 m_i 的确定如下：

$$\begin{cases} \sum_{i=1}^{k} m_i = 1 \\ m_i = \dfrac{C_i}{\sum_{i=1}^{k} C_i} \end{cases} \tag{7}$$

根据以上公式，最终计算出当前 pod 的 CLR，该 CLR 指标在简化计算复杂度和保证应用服务质量的前提下，全面反映出当前 pod 节点整体负载水平。

3.2　ARIMA-Kalman 负载预测模型

3.2.1　ARIMA 预测模型

差分自回归平稳滑动平均模型（ARIMA）是时间序列预测方法，是博克斯和詹金斯提出的一种基于时间序列预测方法。该模型将非平稳时间序列转化为平稳时间序列，让因变量对其滞后值及随机误差项的现值和滞后值进行回归[12]。ARIMA (p,d,q) 模型的数学表达式如下：

$$\phi(B)(1-B)^d x_t = \theta(B)\alpha_t \tag{8}$$

其中 p 为自回归阶数，d 为差分次数，q 为移动平均阶数。B 为后移差分算子，x_t 为原序列，α_t 为均值为 0，方差为 δ^2 的正态白噪声序列。$\phi(B)$ 和 $\theta(B)$ 计算方法如下：

$$\begin{cases} \phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \ldots - \phi_p B^p \\ \theta(B) = 1 - \theta_1 B - \theta_2 B^2 - \ldots - \theta_q B^q \end{cases} \tag{9}$$

$\phi(B)$ 为 p 阶 AR 多项式，$\phi_i (i = 1, 2, \cdots, p)$ 为该多项式的待估系数；$\theta(B)$ 为 q 阶 MA 多项式。$\theta_j (j = 1, 2, \cdots, q)$ 为该多项式的待估系数。

ARIMA 模型的优点在于只需要有限的样本序列，就可以建立精度较高的预测模型。但是该模型存在低阶模型预测精度低，高阶模型参数估计难度大的缺点[13]。

3.2.2　Kalman 滤波算法

Kalman 滤波是一种获取变量的最佳估计，将过去的测量估计误差合并到新的测量误差中来估计将来的误差，来对系统状态进行最优估计的算法[14]。其线性离散的 Kalman 滤波方程如下：

$$x_{k+1} = A_{k+1|k} x_k + w_k \tag{10}$$

$$z_{k+1} = H_{k+1} x_{k+1} + v_{k+1} \tag{11}$$

其中 x_k 代表 t_k 时刻的系统状态向量；z_{k+1} 代表 t_{k+1} 时刻的系统观测向量；A 为转移矩阵，H 为观测矩阵；w_k 和 v_{k+1} 分别表示系统测量过程中的高斯白噪声，协方差分别为 Q_k 和 R_{k+1}。x_k 的最优估计 $\hat{x}_k$ 可按照如下的 Kalman 基本预测递推方程求解[15]。

$$\hat{x}_k = \hat{x}_{k|k-1} + K_k [z_k - H_k \hat{x}_{k|k-1}] \tag{12}$$

$$P_k = [I - K_k H_k] P_{k|k-1} \tag{13}$$

$$\hat{x}_{k|k-1} = A_{k|k-1} \hat{x}_{k-1} \tag{14}$$

$$P_{k|k-1} = A_{k|k-1} P_{k-1} A_{k|k-1}^T + Q_{k-1} \tag{15}$$

$$K_k = P_{k|k-1} H_k^T [H_k P_{k|k-1} H_k + R_k]^{-1} \tag{16}$$

其中 $\hat{x}_{k|k-1}$ 为从 t_{k-1} 时刻对 t_k 时刻的预测值；K_k 为 t_k 时刻的卡尔曼增益矩阵；P_k 是 t_k 时刻 x_k 的状态估计协方差矩阵。

卡尔曼滤波算法采用递归的形式，不需要全部的数据，只需要根据 t_k 时刻的测量值修正 t_{k-1} 时刻的估计值，具有动态加权修正的特性[16]，具有较好的预测精确度。但是 Kalman 滤波算法需要状态方程和测量方程才能保证有很好的预测精度。

3.3　ARIMA-Kalman 预测模型

针对以上对 ARIMA 预测模型和 Kalman 滤波模型的描述，可以看出这两种预测模型都存在不足。本文通过使用 ARIMA-Kalman 算法，将两种预测模型结合。利用 ARIMA 模型建立低阶预测模型，将该低阶模型处理后计算 Kalman 滤波模型的状态方程和测量方程，使用 Kalman 迭代方程进行预测。

对于CLR序列集合 $X(t)$，令 $x_1(t)=x(t)$，$x_2(t)=x(t-1)$，…，$x_p(t)=x(t-p+1)$；$\alpha_1(t)=\alpha(t)$，$\alpha_2(t)=\alpha_1(t-1)$，…，$\alpha_q(t)=\alpha_{q-1}(t-1)$。则ARIMA预测模型如下：

$$\begin{aligned}x(t+1)=&\phi_1(t)x_1(t)+\phi_2(t)x_2(t)+\cdots+\\&\phi_p(t)x_p(t)+\alpha_1(t+1)-\theta_1(t)\alpha_1(t)-\\&\theta_2(t)\alpha_2(t)-\cdots-\theta_q(t)\alpha_q(t)\end{aligned}\tag{17}$$

利用式(10)将上述表达式转为矩阵的形式,则能得到Kalman状态方程：

$$\begin{bmatrix}x_1(t+1)\\x_2(t+1)\\\vdots\\x_p(t+1)\end{bmatrix}=\begin{bmatrix}\phi_1(t)&\phi_2(t)&\cdots&\phi_p(t)\\0&1&\cdots&0\\\vdots&\vdots&\vdots&\vdots\\0&0&\cdots&1\end{bmatrix}\begin{bmatrix}x_1(t)\\x_2(t)\\\vdots\\x_p(t)\end{bmatrix}+\begin{bmatrix}1\\0\\\vdots\\0\end{bmatrix}\alpha_1(t+1)+\begin{bmatrix}-\theta_1(t)\\0\\\vdots\\0\end{bmatrix}\alpha_1(t)+\cdots+\begin{bmatrix}-\theta_q(t)\\0\\\vdots\\0\end{bmatrix}\alpha_q(t)\tag{18}$$

则Kalman测量方程为

$$\begin{aligned}z(t+1)=&[1,0,\cdots,0][x_1(t+1),\\&x_2(t+1),\cdots,x_p(t+1)]^T\end{aligned}\tag{19}$$

该组合算法的具体步骤如下所示。

1)数据采集

利用工具对各个指标进行采集并处理,按照上文中综合指标CLR计算方法对数据进行计算,得出时间序列 $X(t)$。

2)平稳化处理

为了保证数据的平稳化,对采集到的数据信息利用平稳性检验方法ADF检验进行判断。如果是非平稳数据,通过差分进行平稳化处理至ADF检验满足要求。

3)ARIMA模型参数确定

ARIMA (p,d,q) 预测模型中，d 代表在数据平稳化的处理过程中差分的次数。p 和 q 确定的方法有多种,本文中使用固定步长,遍历求出最小AIC信息准则时 p 和 q 的值,从而确定ARIMA模型的参数。

4)Kalman滤波模型预测

根据ARIMA建模求得AR模型和MA模型的待估系数,确定Kalman系统的状态方程和测量方程。利用式(12)~(16)递推求出系统最新估计值。

4 实验

4.1 实验环境

为了验证本文提出的改进的弹性伸缩策略,搭建了两个相同的Kubernetes集群环境,Kubernetes的版本都是1.16.2,每个集群包含一个Master节点和两个Slave节点。其中一个集群使用内置的弹性伸缩方法,另一个集群使用本文提出的弹性伸缩策略。集群的详细配置如表1所示。

表1 实验环境配置

节点	操作系统	处理器	内存	磁盘大小
Master	Centos7.7	4核	4G	80G
Slave	Centos7.7	4核	4G	80G

4.2 实验数据

本文的实验数据使用2018年阿里巴巴公开的生产环境采集的部分容器资源使用率信息。该数据包括CPU利用率、内存使用率,网络以及磁盘IO等多个维度的资源使用情况。由于该数据集数据量庞大,本文选取出同时使用两种及两种以上维度的资源维度的容器数据作为算法计算的基础数据。由于该公开的数据采样的时间间隔不相等,去除时间间隔较近的点并对个别时间间隔较长的数据中间使用均值补差进行补充,使时间间隔大致相等。

4.3 实验步骤及结果分析

本文提出的改进的Kubernetes的伸缩策略需要从功能性和准确性两个方面进行实验验证。所以实验应该包含两个部分:1)使用JMeter工具对Kubernetes的pod进行压力测试,分别记录两个集群的pod的数量变化情况并进行对比。验证本文的伸缩策略在应对负载变化时的预测伸缩效果。2)使用公开容器负载信息,计算出CLR序列。分别使用指数平滑法,ARIMA预测模型和ARIMA-Kalman预测模型进行预测并对这三种预测模型的预测精度进行评估。

首先在两个实验环境中分别部署一个相同的Web应用,伸缩阈值设置为60%,容忍度设置为默认的0.1。利用JMeter工具进行模拟并发访问请求,每隔1min增加并发请求数量并检查当前pod的数量。其中使用本文伸缩策略的实验环境使用Prometheus获取pod的资源信息并计算CLR进行预测。具体数量变化情况如图2所示。

从图2中可以看出,相比内置的伸缩策略方法,本文使用的方法可以提前针对负载的变化趋势进行预测式的弹性伸缩,从而解决Kubernetes内置的伸缩策略的响应延迟问题,保障了服务的质量。

为了比较本文使用的ARIMA-Kalman预测模型和其他模型的精度,使用阿里巴巴的生产环境容

器公开数据集的负载信息数据作为基础数据。简单处理后计算出CLR,将CLR作为模型预测的输入。分别使用绝对误差(MAE)、平均绝对误差(MSE)、平均绝对均方误差(MSRE)这三种最常见的衡量指标对文献[5]中使用的指数平滑法以及文献[6]使用的ARIMA预测模型和本文使用的ARIMA-Kalman预测模型进行评估。

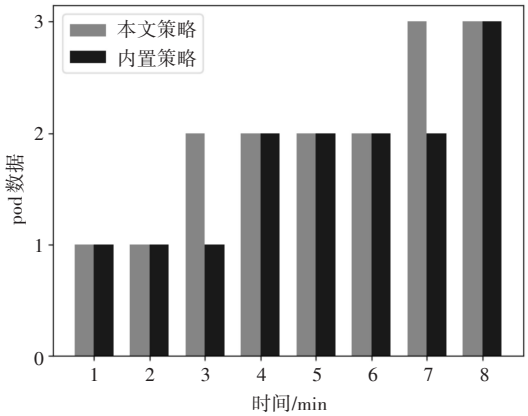

图2　pod数量变化图

每个容器的处理后的采样数据约为650~800个,统一选取前550个信息节点作为训练数据,后100个数据作为预测。指数平滑法,ARIMA预测模型和ARIMA-Kalman的预测误差图已经分别如图3所示。

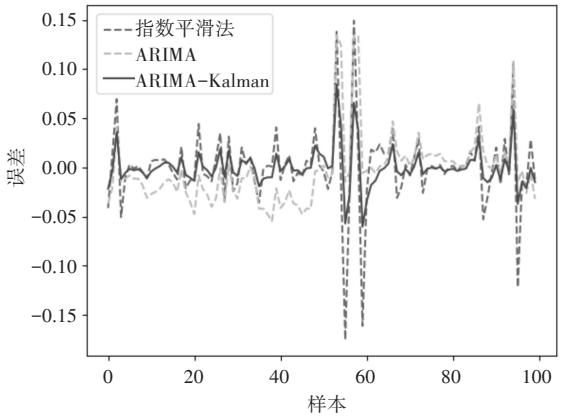

图3　模型预测误差比较图

由图3可以直观看出,ARIMA-Kalman的误差相对其他两种方法更小。由于ARIMA模型的参数确定过程计算量较大,不适合对数据模型进行动态更新,只适合做短期预测,本文一次性预测100个数据点,所以整体误差较大。指数平滑法由于使用了所有历史节点的信息,在使用中会消耗更多的内存资源,而且没有考虑到数据变化的规律性。ARIMA-Kalman模型通过建立低阶的ARIMA模型确定状态转移方程,使用Kalman模型进行更新迭代预估,其精确度相比ARIMA和指数平滑法较大的提升。表2是对这三种模型的评估。

表2　各模型评价指标对比

节点	MAE	MSE	RMSE
指数平滑法	0.02269	0.00185	0.04301
ARIMA	0.02392	0.00135	0.03681
ARIMA-Kalman	0.01208	0.00038	0.01959

可以看出,相比于指数平滑法、ARIMA模型,本文所使用的ARIMA-Kalman预测模型的预测精度更优。在面对负载变化的情况,能够准确地进行预测。

5　结语

本文针对Kubernetes内置的弹性伸缩机制的单一衡量指标和响应延迟的问题,提出了一种改进的弹性伸缩策略。该策略通过定义统一负载评价指标CLR,有效衡量出复杂应用的整体负载水平。同时通过上下限阈值的设置,简化了计算的复杂度并且保证了服务的质量。使用ARIMA-Kalman预测模型对应用当前CLR进行预测,提前伸缩以保证服务质量。实验证明,该预测模型相比于指数平滑法和ARIMA预测模型具有更高的精度。

参 考 文 献

[1] Merkel D. Docker: Lightweight Linux Containers for Consistent Development and Deployment[J]. Linux J., Houston, TX: Belltown Media, 2014(239):76-90.

[2] 董博,王雪,索菲,等. 基于Docker的虚拟化技术研究[J]. 辽宁大学学报(自然科学版),2016,43(04):327-330.
DONG Bo, WANG Xue, SUO Fei, et al. Research on Virtualization Technology Based on Docker[J]. Journal of Liaoning University (Natural Science Edition), 2016, 43(04): 327-330.

[3] 武志学. 云计算虚拟化技术的发展与趋势[J]. 计算机应用,2017,37(04):915-923.
WU Zhixue. Advances on Virtualization Technology of Cloud Computing[J]. Computer Application, 2017, 37(04): 915-923.

[4] Bernstein D. Containers and Cloud: From LXC to Docker to Kubernetes[J]. Cloud Computing, IEEE, 2014, 1(3): 81-84.

[5] Burns B, Grant B, Oppenheimer D, et al. Borg, Omega, and Kubernetes[J]. Communications of the ACM, 2016, 59(5):50-57.

[6] 张可颖, 彭丽苹, 吕晓丹, 等. 开源云上的Kubernetes

(下转第372页)

(5):21-23.

[13] 刘耀芳,纪彦东. 基于BP网络的往复压缩机气缸故障诊断技术[J]. 粮油加工,2010(10):143-146.
LIU Yaofang, JI Yandong. Cylinder Fault Diagnosis Technology of Reciprocating Compressor Based on BP Network [J]. Grain and oil Processing, 2010 (10): 143-146.

[14] 刘凡平. 神经网络与深度学习应用实战[M]. 北京:电子工业出版社,2018.
LIU Fanping. Practical Application of Neural Network and Deep Learning [M]. Beijing: Electronic Industry Press, 2018.

[15] 曾慧浩,郭建胜. 双向LSTM神经网络的航空发动机故障预测[J]. 空军工程大学学报,2019,20(4):26-32.
ZENG Huihao, GUO Jiansheng. Aeroengine Fault Prediction Using Bidirectional LSTM Neural Network [J]. Journal of Air Force Engineering University, 2019, 20 (4):26-32.

(上接第331页)

弹性调度[J]. 计算机技术与发展, 2019, 29(02): 115-120.
ZHANG Keying, PENG Liping, LV Xiaodan, et al. Elastic Scheduling Strategy for Private Cloud Resource Based on Kubernetes and Openstack [J]. Computer Technology and Development, 2019,29(02): 115-120.

[7] 谢文舟,孙艳霞. 基于Kubernetes负载特征的资源预测模型研究[J]. 网络安全技术与应用,2018(04):27-28.
XIE Wenzhou, SUN Yanxia. Research on Resource Prediction Model Based on Kubernetes Load Characteristics [J]. Network Security Technology and Application, 2018 (04): 27-28.

[8] 陈雁,黄嘉鑫. 基于Kubernetes应用的弹性伸缩策略[J]. 计算机系统应用,2019,28(10):213-218.
CHEN Yan, HUANG Jiaxin. Elastic Scaling Strategy Based on Kubernetes Application [J]. Computer Systems & Applications, 2019,28 (10): 213-218.

[9] 杨茂,陈莉君. 基于Kubernetes的容器自动伸缩技术的研究[J]. 计算机与数字工程, 2019, 47 (09): 2217-2220,2232.
YANG Mao, CHEN Lijun. Research on Container Auto-scaling Based on Kubernetes [J]. Computer & Digital Engineering, 2019,47 (09): 2217-2220,2232.

[10] 马小淋. 一种基于负载特征预测的容器云弹性伸缩策略[J]. 信息安全研究, 2019, 5(03):54-59.
MA Xiaolin. A Container Cloud Elastic Scaling Strategy Based on Load Characteristics Prediction [J]. Information Security Research, 2019,5(03): 54-59.

[11] 屠雪真,杨海潮. 基于Kubernetes的水平弹性扩缩容系统[J]. 计算机与现代化,2019(07):25-31.
TU Xuezhen, YANG Haichao. A Horizontal Elastic Expansion and Contraction System Based on Kubernetes [J]. Computer and Modernization, 2019(07): 25-31.

[12] 王燕. 应用时间序列分析[M]. 北京: 中国人民大学出版社,2005: 146-149.
WANG Yan. Applied Time Series Analysis [M]. Beijing: China Renmin University Press,2005: 146-149.

[13] 潘迪夫,刘辉,李燕飞. 基于时间序列分析和卡尔曼滤波算法的风电场风速预测优化模型[J]. 电网技术, 2008(07):82-86.
PAN Difu, LIU Hui, LI Yanfei. A Wind Speed Forecasting Optimization Model for Wind Farms Based on Time Series Analysis and Kalman Filter Algorithm [J]. Grid technology, 2008 (07): 82-86.

[14] 谷建伟,隋顾磊,李志涛,等. 基于ARIMA-Kalman滤波器数据挖掘模型的油井产量预测[J]. 深圳大学学报(理工版),2018,35(06):575-581.
GU Jianwei, SUI Gulei, LI Zhitao, et al. Oil well Production Forecasting Method Based on ARIMA-Kalman Filter Data Mining Model [J]. Journal of Shenzhen University (Science and Technology Edition), 2018, 35 (06): 575-581

[15] 薛洋,杨光,许雷. 基于Kalman-ARIMA模型的大坝变形预测[J]. 中国农村水利水电,2016(12):117-119, 123.
XUE Yang, YANG Guang, XU Lei. A Prediction of Dam Deformation Based on Kalman-ARIMA Model [J]. China Rural Water Conservancy and Hydropower, 2016 (12): 117-119,123.

[16] 徐洪俊, 张锦东, 张其林. 基于时间序列模型和扩展卡尔曼滤波算法的结构响应预测[C]// 第十九届全国现代结构工程学术研讨会论文集,2019:389-393.
XU Hongjun, ZHANG Jindong, ZHANG Qilin. Structural Response Prediction Based on the Auto Regressive Moving Model and Extended Kalman Filter [C]// Proceedings of the 19th National Symposium on Modern Structural Engineering,2019:389-393.