DOI:10.19651/j.cnki.emt.2518053

基于 WTT-iTransformer 时序预测的容器群伸缩策略研究*

陈奇超 叶 楠 曹炳尧
(上海大学特种光纤与光接入网重点实验室 上海 200444)

摘 要:Kubernetes 默认的 HPA 策略因其特有的响应性机制而存在扩缩容滞后的局限。为了提高资源的响应性能和资源利用率,本文引入了基于时序资源负载预测的弹性伸缩策略,预测部分创新得提出了 WTT-iTransformer 模型对集群资源进行预测。已知 iTransformer 不仅在长期序列预测表现优异,还可通过变量序列作为 token 嵌入获取了多变量间的关联性。本文通过增加了小波变换卷积层 WTConv2d 和多尺度时间卷积网络的 WTT-iTransformer 模型可以更精确地从时、频域两方面提取资源时间序列的长期特征与依赖关系,更符合容器使用特征的预测。基于该模型的负载变化预测,能够实现高、低流量发生的初期进行快速扩缩容,以解决反应滞后和资源利用率低的问题。实验结果表明,WTT-iTransformer 在训练过程中表现出更好的稳定性和更低的训练误差,能够较为准确地预测集群负载的变化趋势,改进的弹性伸缩策略与 Kubernetes 传统的 HPA 相比更加智能、稳定,在负载特征明显、突发性负载较多的场景展现出显著提升,具有广泛的应用潜力。

关键词:Kubernetes;时序预测模型 WTT-iTransformer;负载预测;混合弹性伸缩策略;小波变换卷积;时间卷积网络;iTransformer 模型

中图分类号:TN915.5 **文献标识码**:A **国家标准学科分类代码**:520.2010

Research on container cluster scaling strategies based on WTT-iTransformer time series prediction

Chen Qichao Ye Nan Cao Bingyao
(Key Laboratory of Specialty Fiber Optics and Optical Access Networks, Shanghai University, Shanghai 200444, China)

Abstract: The default Horizontal Pod Autoscaler strategy in Kubernetes has limitations due to its inherent response mechanism, leading to scaling delays. To improve resource response performance and resource utilization, this paper introduces an elastic scaling strategy based on time-series resource load prediction. The proposed prediction model, WTT-iTransformer, is specifically designed to forecast cluster resources. It is known that iTransformer excels in long-term sequence prediction and can capture the correlations between multiple variables by embedding variable sequences as tokens. By adding a Wavelet Transform Convolutional layer and integrating a Multi-Scale Temporal Convolutional Network, the WTT-iTransformer model is constructed, enabling more precise extraction of long-term features and dependencies in resource time-series from both the time and frequency domains, which aligns better with the prediction of container usage characteristics. Based on the load variation prediction of this model, rapid scaling can be implemented at the early stages of high and low traffic occurrences, addressing the issues of delayed responses and low resource utilization. Experimental results show that the WTT-iTransformer demonstrates better stability and lower training error during training, accurately predicting cluster load trends. The improved elastic scaling strategy, compared to the traditional HPA in Kubernetes, is more intelligent and stable, showing significant improvements in scenarios with notable load characteristics and frequent burst traffic, thus holding broad application potential.

Keywords: Kubernetes; WTT-iTransformer time-series prediction model; load prediction; hybrid elastic scaling strategy; wavelet transform convolution; temporal convolution network; iTransformer model

0 引 言

随着容器编排技术的快速发展,Kubernetes(K8s)凭借其成熟性与灵活性脱颖而出[1]。K8s 以 Pod 为最小调度单元,并通过 horizontal Pod autoscaler(HPA)实现基于负载的自动扩展。尽管 HPA 可以依据预设阈值动态调整 Pod 副本,但其被动响应机制存在一定滞后性,可能导致服务质量下降。此外,在面对流量突发变化时,HPA 的反应能力

收稿日期:2025-02-11
* 基金项目:国家重点研发计划(2021YFB2900800)、高等学校学科创新引智计划(111 项目)(D20031)资助

也相对不足。因此学者们通常聚焦于改进原生策略和引入更为高效的负载预测模型来实施改进。

原生弹性伸缩策略的优化方向主要聚焦于阈值动态化与响应效率提升。针对传统预设阈值方案的局限性,学界提出了多种动态调控机制。例如,Al-Sharif 团队[2]设计的智能资源调度系统通过实时监控集群运行指标实现阈值动态校准,显著提升了资源调配的灵活性。在保障系统稳定性方面,陈雁等[3]创新性地引入自适应步长调整机制,其研究成果显示服务失败率可降低 23.6%。单朋荣等[4]通过融合多维监控指标与智能伸缩算法,在容器云平台中实现了资源利用率与服务质量的双重优化。

然而,基于 K8s 本身调度策略的改良存在不可避免的滞后性问题,基于负载预测的主动伸缩模型成为研究热点。石硕等[5]构建的双层 LSTM 预测框架,通过分析多维负载特征实现了综合负载的时序预测。在容器动态调度领域,陈烨[6]创新应用门控循环单元改进预测模型,实现了 Web 应用 Pod 数量的精准预估。在预测算法方面,Farahnakian 团队[7]采用线性回归构建 CPU 利用率预测模型;Calheiros 团队[8]则基于 ARIMA 时间序列分析开发云工作负载预测工具。此外,Janardhanan 等[9]将注意力机制融入 LSTM 模型,有效提升了数据中心 CPU 使用率的预测精度。而在混合模型研究方面,覃盛[10]提出的 ARIMA-LSTM 组合预测器在 HPA 策略中展现出优越性能;Bi 等[11]开发的 SGW-S 组合模型在分布式云服务场景中可以有效预测下一时间段的工作负载。

在深入学习相关领域内前人所做的工作和研究后,本文主要关注于预测模型的改进提升与弹性伸缩策略的优化研究。一方面基于 iTransformer 模型,提出了新时序预测模型 WTT-iTransformer 来对 K8s 集群资源进行预测;另一方面,设计了基于 WTT-iTransformer 模型的混合弹性伸缩策略,通过预测集群资源未来使用情况并结合实时负载状态来优化调度,从而进一步提升资源管理效率。

1 预测模型 WTT-iTransformer

本文基于 iTransformer 提出了一种改进的时间序列预测模型 WTT-iTransformer。该模型基于现有的 Transformer 变体 iTransformer,在网络中引入 WTConv2d 小波变换卷积层和多尺度 TCN 时间卷积层,再将每个序列独立嵌入到变量 token 中,并应用 iTransformer 的注意力机制和共享前馈网络来分别捕捉序列的长短期变量依赖关系和周期性和实现序列表征,从而改善了模型在时、频域中提取复杂时间序列变量特征相关性的能力,提高了模型预测的准确性。

1.1 小波变换卷积

本文所针对的小波变换指 Haar 小波变换,因为它高效且简单,(其他的小波基底也可以使用,但是计算成本会有所改变)并且由于本文所涉及的时间序列在每一个时间点包含多个变量值(即每个时间点是一个多维向量),在这种情况下,二维小波变换是一个非常合适的工具。

若给定序列输入 $\boldsymbol{X}$,在第一维度时间上的一级 Haar 小波变换是通过与核 $[1,1]/\sqrt{2}$ 和 $[1,-1]/\sqrt{2}$ 进行深度卷积,然后应用一个标准的 2 倍下采样算子来实现的。为了执行二维 Haar 小波变换,需要在两个维度上组合该操作,使用以下 4 个滤波器集合以步长为 2 进行深度卷积,如式(1)所示。

$$\boldsymbol{f}_{LL}=\frac{1}{2}\begin{bmatrix}1&1\\1&1\end{bmatrix},\boldsymbol{f}_{LH}=\frac{1}{2}\begin{bmatrix}1&-1\\1&-1\end{bmatrix},\boldsymbol{f}_{HL}=\frac{1}{2}\begin{bmatrix}1&1\\-1&-1\end{bmatrix},\boldsymbol{f}_{HH}=\frac{1}{2}\begin{bmatrix}1&-1\\-1&1\end{bmatrix} \tag{1}$$

式中:$\boldsymbol{f}_{LL}$ 是一个低通滤波器,而 $\boldsymbol{f}_{LH}$、$\boldsymbol{f}_{HL}$、$\boldsymbol{f}_{HH}$ 是一组高通滤波器。对于每一个输入通道,卷积的输出如式(2)所示。

$$[\boldsymbol{X}_{LL},\boldsymbol{X}_{LH},\boldsymbol{X}_{HL},\boldsymbol{X}_{HH}]=Conv([\boldsymbol{f}_{LL},\boldsymbol{f}_{LH},\boldsymbol{f}_{HL},\boldsymbol{f}_{HH}],\boldsymbol{X}) \tag{2}$$

卷积后的输出有 4 个通道,每个通道(在每个空间维度上)的分辨率都是 $\boldsymbol{X}$ 的一半。$\boldsymbol{X}_{LL}$ 是 $\boldsymbol{X}$ 的低频分量,而 $\boldsymbol{X}_{LH}$、$\boldsymbol{X}_{HL}$、$\boldsymbol{X}_{HH}$ 分别是其水平、垂直和对角高频分量。

由于上述滤波器核构成了一个正交归一化基,因此应用逆小波变换(IWT)可以通过转置卷积获得,如式(3)所示。

$$\boldsymbol{X}=Conv-transposed([\boldsymbol{f}_{LL},\boldsymbol{f}_{LH},\boldsymbol{f}_{HL},\boldsymbol{f}_{HH}],[\boldsymbol{X}_{LL},\boldsymbol{X}_{LH},\boldsymbol{X}_{HL},\boldsymbol{X}_{HH}]) \tag{3}$$

然后通过递归分解低频分量来获得级联小波分解,每一级的分解由式(4)给出:

$$\boldsymbol{X}_{LL}^{(i)},\boldsymbol{X}_{LH}^{(i)},\boldsymbol{X}_{HL}^{(i)},\boldsymbol{X}_{HH}^{(i)}=WT(\boldsymbol{X}_{LL}^{(i-1)}) \tag{4}$$

式中:$\boldsymbol{X}_{LL}^{(0)}=\boldsymbol{X}$,$i$ 是当前级别,这会导致低频分量的频率分辨率增加,空间分辨率降低。

引入小波变换卷积层至网络中有两个主要的技术优势。一方面,小波变换的每一级都会增加层的感受野大小,而可训练参数的数量只会有小幅增加。第二个好处是,小波变换卷积层的设计比标准卷积更能捕捉低频信息。这是因为输入的低频信息经过重复的小波变换分解后,低频信息被强调,从而增加了层对低频信息的响应。本文实验中应用的小波变换卷积过程如图 1 所示。

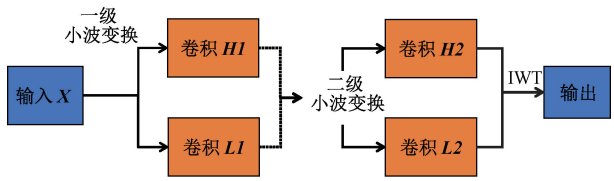

图 1 小波变换卷积流程

Fig. 1 Process of the WTConv2d layer

1.2 时间卷积网络

时间卷积网络(temporal convolutional network, TCN)广泛应用于时间序列预测。相较于传统的循环神经

网络，TCN 在捕捉时序依赖关系方面表现出明显优势，能够实现并行计算、高效处理长时间依赖问题。本文应用的 TCN 网络整体架构如图 2 所示。

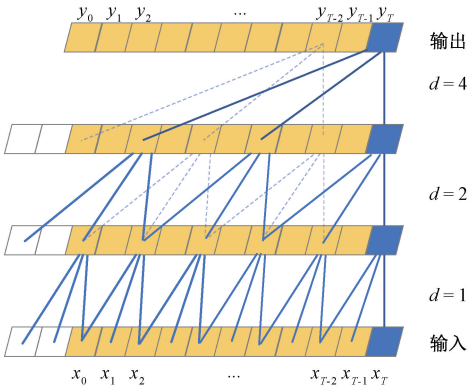

图 2　TCN 网络整体架构

Fig. 2　The overall architecture of TCN

图 2 左侧所示为膨胀因果卷积，其通过引入间隔采样机制扩展了传统卷积操作，这种周期性插入零值间隙的采样策略能够使网络捕获长距离时序依赖，同时保持因果约束条件，并可以有效地扩大卷积操作的感受范围。其核心参数是扩张因子，决定了卷积核中相邻元素之间的间距。扩张因果卷积的数学表示如式(5)所示。

$$y(t)=\sum_{i=0}^{k-1} f(i)\cdot x(t-d\cdot i) \tag{5}$$

式中：f 是卷积核，k 是卷积核大小，x 是输入，d 是扩张率。

图 2 右侧部分是 1×1 卷积构成的残差连接，两条链路并行，构成残差块，每个残差块包含：一维膨胀因果卷积层，权重归一化层，ReLU 激活函数层，Dropout 层。残差块再串联构成时序卷积神经网络。具体而言，假设输入为 x，经过若干层计算后得到 $F(x)$，则残差块的最终的输出不是单独的 $F(x)$，而是 $x+F(x)$，即输入和输出的加和。这种结构使得梯度在反向传播过程中能够直接流向更早的层，从而有效减少梯度消失的问题。

1.3　iTransformer

1)iTransformer 特点

近年来 Transformer 及其变体被用于预测的热情逐渐升高，如朱彦民等[12]通过引入线性解码器结构构建 Transformer 模型，并使用该模型对风电机组故障进行预测研究。李欣宇等[13]通过 Transformer 模块强化了区域特征的关联性，大大强化了有用信息的表达。模型变体中，朱焕宇等[14]提出一种基于 Swin Transformer 和注意力特征金字塔的子午线轮胎缺陷分割算法 Swin DAA。黄星华等[15]提出使用 Transformer 的变体 VOLO 构造特征提取器以获取细粒度更佳的故障特征表示。赵昱坡等[16]为了避免单一模型预测的局限性，采用 LSTM 模型改进了 Transformer 结构中的解码器。这些预测器利用 Transformer 对时间序列的时间 token 进行全局依赖性建模，每个 token 由相同时间戳的多个变体组成。

然而，Transformer 在预测多维数据序列时面临挑战。此外，每个时间 token 的嵌入融合了代表潜在延迟事件和不同物理测量的多个变量，这可能导致无意义的注意力特征。基于以上原因，本文选择采用 iTransformer 模型。iTransformer 模型在不对任何 Transformer 模型基本组件进行任何修改的情况下重新利用了 Transformer 架构，将每个变量的整个时间序列作为一个变量 token，这样的 token 聚合了变量的完整序列，可以更加以变量为中心。具体而言，以往时序预测模型的主流做法是将输入变量的同一时刻表示为词来获得以时间为单位的词序列，每个特征表示只能反映一个时间戳，感受野小，使得模型在处理长序列输入时计算复杂度提高，预测效果可能会下降；而在 iTransformer 中，变量的整条序列被独立地映射为变量词，然后其以变量为主体，再通过注意力机制即可自然地挖掘以词为单位的多变量关联。此外，iTransformer 的前馈网络和层归一化互相配合，可以在时间维上逐层编码历史观测特征的同时，有效消弭变量测量单位之间的范围差异。两种模型的 token 视角对比如图 3 所示。

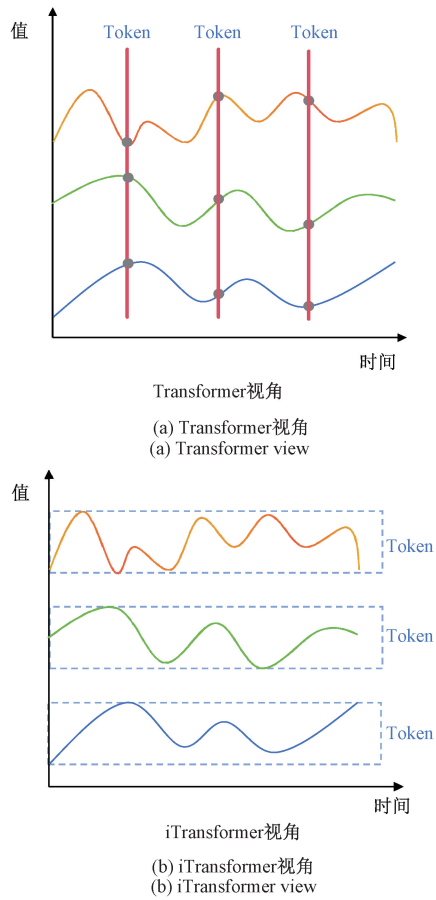

图 3　Transformer 与 iTransformer 模型间 token 对比

Fig. 3　Token comparison between Transformer and iTransformer models

2)iTransformer 整体架构

iTransformer 结构如图 4 所示,其采用 Transformer 的纯编码器架构,包括嵌入、投影和 Transformer 模块。

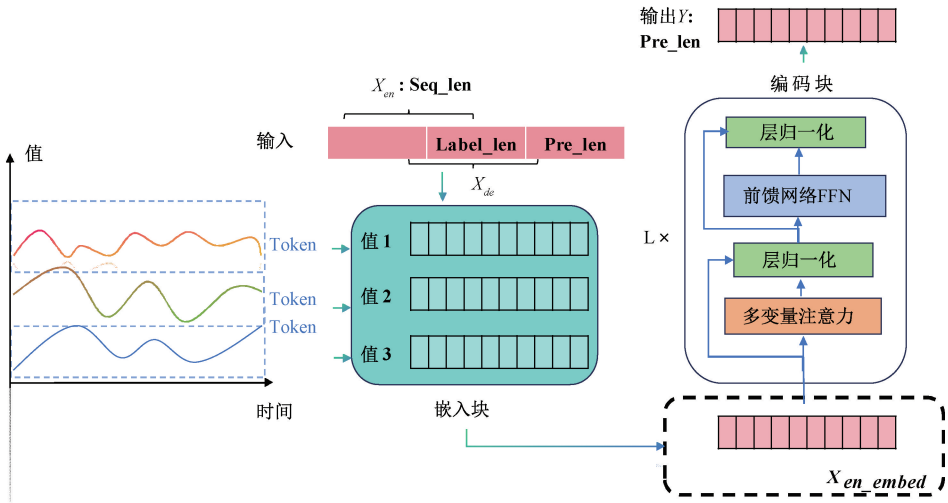

图 4 iTransformer 的整体结构

Fig. 4 The overall architecture of iTransformer

设输入输出如式(6)所示。

$$\begin{cases}\boldsymbol{X}=\{x_1,\cdots,x_T\}\in\mathbf{R}^{T\times N}\\ \boldsymbol{Y}=\{x_{T+1},\cdots,x_{T+S}\}\end{cases}\tag{6}$$

式中:历史序列大小为 $T\times N$,T 为时间序列长度,N 为特征维度。S 为预测序列长度。

中间通过嵌入层,编码块的多层堆叠以及最后的输出处理来实现对未来长度为 S 时间序列的预测。iTransformer 编码块主要由层归一化,前馈网络和自注意力模块组成的 L 个模块堆叠组成。在 iTransformer 中,层归一化被应用于单变量序列中,所有序列都会归一化为高斯分布,因此可以减少由测量引起的差异;前馈网络利用的是每个变量 token 的序列表示,根据通用近似定理,它们可以提取复杂的表示来描述时间序列,通过转置模块的堆叠对观察到的时间序列进行编码,并利用密集的非线性连接对未来序列的表示进行解码;自注意力模块在该模型中用于建模不同变量的相关性,iTransformer 模型将一个单变量的整个序列视为独立过程,通过自注意力模块全面提取时间序列表示,采用线性投影获取 $\boldsymbol{Q}$、$\boldsymbol{K}$、$\boldsymbol{V}$(查询,键,值)的值,计算前 Softmax 分数,揭示变量之间的相关性。$\boldsymbol{Q}$、$\boldsymbol{K}$、$\boldsymbol{V}$ 通过自注意力模块采用线性投影得到,d_k 为投影维数。q_i,k_j 属于同一个变量 token 的特定 query 和 key,softmax 前的每一项得分都被表述为式(7):

$$\boldsymbol{A}_{i,j}=(\boldsymbol{QK}^{\mathrm{T}}/\sqrt{d_k})_{i,j}\propto q_i^{\mathrm{T}}k_j\tag{7}$$

由于每个 token 之前都在其特征维度上进行了归一化处理,因此每一项可以一定程度上揭示变量间的相关性,整个分数图 $\boldsymbol{A}$ 显示了配对变量 token 之间的多变量相关性。

1.4 WTT-iTransformer 整体架构

针对 Transformer 在捕捉基本序列表征和刻画多变量相关性方面的局限,本文提出了一种基于 iTransformer 的改进时间序列预测算法模型 WTT-iTransformer,以对 K8s 集群资源负载进行预测。

该模型添加 WTConv2d 小波变换卷积层提取训练数据集的隐藏频域特征,增强模型对频域上多变量相关性及其周期性的发现能力;然后取代原有的嵌入层并集成 2、4、6 尺寸的多尺度的 TCN 卷积块捕捉训练集数据中的时间域依赖关系,并将输出的特征进行融合;然后再基于编码器,将每个序列独立嵌入到变量 token 中,应用注意力机制和共享前馈网络来分别捕捉序列的长短期变量依赖关系和周期性和实现序列表征;最后通过投影映射层将每个 token 投影映射为预测结果得到输出。WTT-iTransformer 的整体架构如图 5 所示。

2 预测算法实验

2.1 训练数据集选取

为全面评估提出的模型的泛化能力与有效性,本文选取电力系统运行数据(ETTH1、ETTh1/2)、设备级时序数据(ETTm2)与阿里巴巴的公开集群数据集 cluster-trace-v2023 中的 machine usage 数据集进行测试,详细信息如表 1 所示。

表 1 中的 cluster-trace-v2023 数据集包含线上 4 000 多台服务器的运行 21 天的真实数据,本实验选择其中一台 Web 应用性质的服务器进行分析,最终数据集如表 2 所示。

选择其中除 CPU 利用率以外的数据列作为输入参数,CPU 利用率为预测项,进行多元参数对 CPU 的预测。数据集原始的数据无法直接使用,需要在实验开始前进行数据清洗,首先进行异常值处理,数据集为各项资源利用

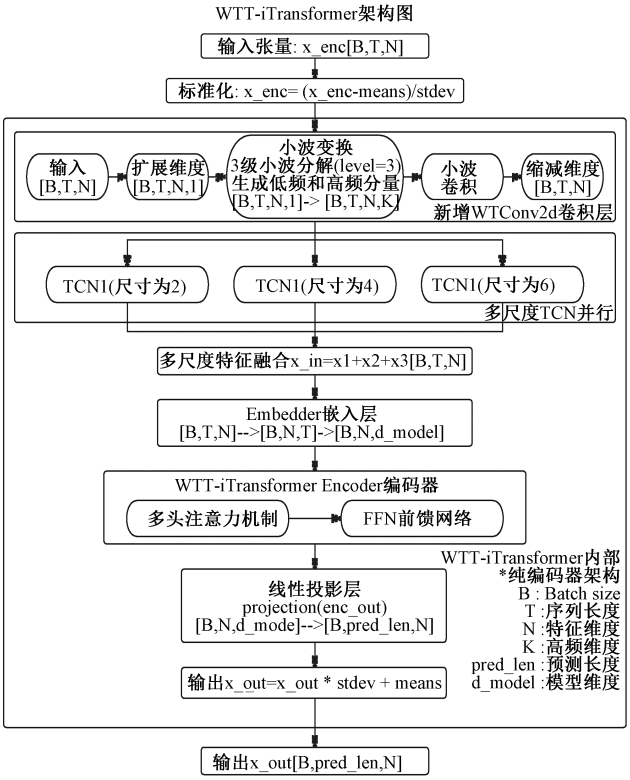

图 5　WTT-iTransformer 整体架构

Fig. 5　WTT-iTransformer overall architecture

表 1　数据集特征

Table 1　Characteristics of the dataset

数据集	数据量/条	时间间隔/s
ETTH1	17 420	3 600
ETTH2	17 420	3 600
ETTm2	69 680	900
cluster-trace-v2023	174 982	10

表 2　数据表参数说明

Table 2　Description of table parameters

参数名	说明
date	时间戳
net_in	入向流量归一化数值([0,100])
net_out	出向流量归一化数值([0,100])
disk_io_percent	磁盘利用率
mem_util_percent	内存利用率
cpu_util_percent	CPU 利用率

率的统计，其中存在超过 100 的值，除去数据中的异常值，其次对缺失特征的数据项删除，防止在后续计算损失函数时出现错误，最后进行数据聚合，由于其采样并非严格均匀，采用线性插值法进行数据处理，然后按照每 10 s 采样一次的频率进行采样。

2.2　实验细节设计

为了验证第一章提出的 WTT-iTransformer 模型的有效性，本文额外进行了对照实验。除改进的基准模型 iTransformer 外，本文还以其他主流的时间序列模型 Transformer、Informer、TimesNet 作为对比模型。WTT-iTransformer 代码部分均基于 Pytorch 2.0.1 框架并使用 Python 3.11.4 编写实现。模型部署在云服务器中的 Windows 10 系统主机上，且 GPU 为单张 RTX4090D。具体参数如表 3 所示。

表 3　WTT-iTransformer 模型主要参数配置

Table 3　Main parameter configuration of WTT-iTransformer

参数名称	参数值
激活函数	RELU
优化器	Adam
初始学习率	0.000 1
损失函数	RMSE
Dropout 系数	0.2
TCN 卷积核数量	3
批处理大小	32

2.3　模型评价标准

为评价模型的准确程度，本文采用预测精度量化双指标：平均绝对误差(MAE)和均根方误差(RMSE)，计算公式为：

$$r_{\mathrm{MAE}}=\frac{1}{n}\sum_{i=1}^{n}|y_i-\hat{y}_i| \tag{8}$$

$$r_{\mathrm{RMSE}}=\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2} \tag{9}$$

式中：n 为样本总量，y_i 为第 i 项实际值，$\hat{y}_i$ 为第 i 项的模型预测值。MAE 反映绝对误差的算术均值，对异常值具有鲁棒性；RMSE 通过平方运算放大显著偏差，更关注极端误差的影响。两指标数值越小表征预测偏差越小，即模型性能越好。

2.4　模型负载预测结果验证

将数据分为 70%的训练集，10%的验证集和 20%的测试集，使用本文的 WTT-iTransformer 进行训练，部分预测结果与真实负载值的对比图如图 6 所示。

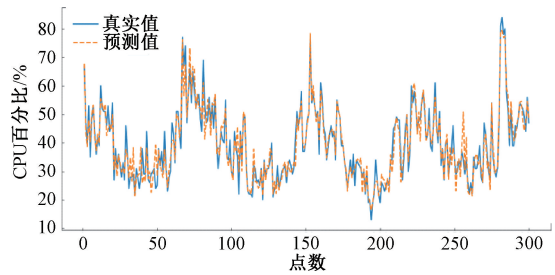

图 6　真实-预测对比图

Fig. 6　Actual-predicted comparison diagram

为进一步验证本文提出的 WTT-iTransformer 模型的准确性,本实验选择与其他模型进行比较。表 4 为 WTT-iTransformer 在与其他对比模型在测试集上的性能评估结果。在数据集 cluster-trace-v2023 中,设置 360/1440 时间步长分别对应 1 h 与 4 h 趋势预测任务。针对公开数据集 ETTm2,采用 96/288 步长对应 24 h 与 72 h 预测;而 ETTh1/ETTh2 数据集则配置 72/168 步长对应 72 h 与 168 h 预测。对比模型及除阿里数据集 cluster-trace-v2023 以外的实验数据均来自项目(2021YFB1715200)所开放的数据平台 Time Series Library。

表 4 不同数据集上 5 种模型的性能表现

Table 4 Performance of five models on different datasets

数据集	预测长度/个	WTT-iTransformer		iTransformer		TimesNet		Transformer		Informer	
		MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
ETTm2	96	**3.052 6**	**4.254 6**	3.211 2	4.427 6	3.587 6	4.921 3	3.845 6	5.287 6	4.247 2	5.821 5
	288	**3.812 5**	**5.312 9**	4.123 5	5.567 2	4.555 4	6.125 6	5.123 2	6.651 2	5.254 3	7.126 4
	平均	3.432 6	4.783 8	3.667 4	4.997 5	4.071 5	5.523 5	4.484 4	5.969 4	4.750 8	6.474 0
ETTh1	72	**3.175 6**	**4.365 9**	3.524 5	4.841 7	4.576 1	6.292 7	4.224 2	5.808 6	3.872 0	5.324 4
	168	**3.968 8**	**5.454 6**	4.416 8	6.058 3	5.723 7	7.865 3	5.122 3	7.265 8	4.841 5	6.655 3
	平均	3.572 2	4.910 3	3.970 7	5.450 0	5.149 9	7.079 0	4.673 3	6.537 2	4.356 8	5.989 9
ETTh2	72	**3.475 5**	**4.783 2**	3.685 6	5.065 3	4.416 6	6.072 4	4.048 2	5.566 5	4.784 3	6.578 3
	168	**4.343 5**	**5.978 8**	4.643 1	6.325 2	5.529 8	7.593 2	5.060 3	6.957 5	5.223 4	8.222 5
	平均	3.909 5	5.381 0	4.164 4	5.695 3	4.973 2	6.832 8	4.554 3	6.262 0	5.003 9	7.400 4
cluster-trace-v2023	360	**2.943 8**	**4.118 8**	3.133 9	4.332 6	3.760 7	5.199 1	3.447 3	4.765 0	4.073 0	5.632 4
	1 440	**3.679 8**	**5.148 5**	3.917 4	5.415 8	4.700 9	6.498 9	4.309 1	5.957 4	5.091 3	7.040 5
	平均	3.311 8	4.633 6	3.525 6	4.874 2	4.230 8	5.849 0	3.878 2	5.361 2	4.582 2	6.336 4

从表 4 数据来看,本文提出的 WTT-iTransformer 模型在 4 种数据集上的综合表现较好,评价指标相对优异。相对于改进的基准模型 iTransformer,MAE 指标降低 4.93%~7.54%,RMSE 指标降低 3.91%~5.57%。

从 Transformer 及其所有变体和 WTT-iTransformer 模型来看,WTT-iTransformer 在评价指标上均有所提升。本文认为这是因为传统的 Transformer 及其所有变体模型均采用了改变组件(注意力机制的复杂程度)的方法来提升模型的预测和提取特征的能力,却忽略了对模型的整体架构进行调整的可能性。WTT-iTransformer 和 iTransformer 创新性地倒置注意力机制和前馈网络的职责,增强了对多变量相关性建模和消除变量间差异的能力。而本文提出的 WTT-iTransformer 模型相较于 iTransformer 而言,还通过引入小波卷积层和多尺度 TCN 层来实现对输入序列的不同频率成分的特征进行提取和融合。总体而言,本文提出的 WTT-iTransformer 模型预测方法在一般的时间序列预测上具有明显的优势。

2.5 增加历史数据长度实验

关于 Transformer 系列模型的以往许多研究已表明,Transformer 模型的预测效果与回溯长度之间并非呈现正相关关系,这可能是由于注意力机制在输入序列增长时的分散现象导致的。因此,基于 iTransformer 的创新架构设计,本文选取 ETTm2 和 ETTh1 作为实验数据集,在固定预测点数长度为 96 的情况下,通过增加历史数据长度来验证 WTT-iTransformer 模型的长序列预测表现。不同回溯长度下的模型表现对比结果如图 7、8 所示。

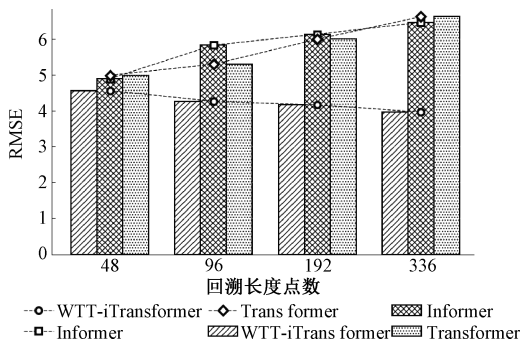

图 7 模型表现对比图(ETTm2)

Fig. 7 Comparison of model performance under different look-back lengths(ETTm2)

根据图 7、8 所示的对比实验结果发现,在回溯长度逐步增加的实验条件下,WTT-iTransformer 模型表现出显著性能优势。这不仅证实了 WTT-iTransformer 架构中前馈网络部分在时序维度上拥有优秀的特征处理能力,可以克服传统 Transformer 系列模型在处理长序列数据时注意力分散的固有缺陷,也验证了其注意力机制部分能够有效提取扩展回溯窗口的多尺度特征信息从而提升预测精度。在历史观测点数增加的情况下,WTT-iTransformer 模型由于注意力机制与前馈网络职责倒置而展现出的预测误差逐步减小的趋势特点,在一定程度上也同样证明了该模

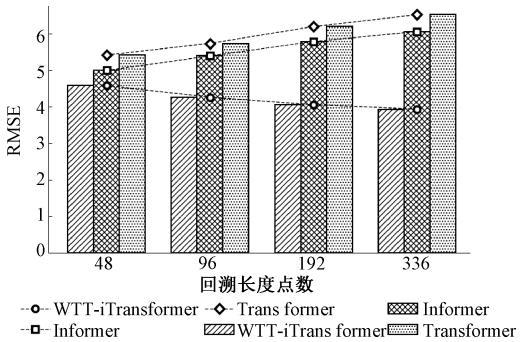

图 8　模型表现对比图(ETTh1)

Fig. 8　Comparison of model performance under different look-back lengths(ETTh1)

型架构的合理性。

2.6　消融实验

消融实验选取阿里数据集 cluster-trace-v2023,且设置预测长度为 360 即 1 h。消融实验结果如表 5 所示。从表 5 中可以看出,WTT-iTransformer 的 MAE 和 RMSE 均优于其他模型,这主要是因为小波变换卷积增强了模型对于获取数据集时频域特征的能力;多尺度 TCN 提取了更加稳定和可靠的特征,大幅提升模型的泛化能力和鲁棒性;iTransformer 提取了多变量之间的相关性,做到了更深层次的特征提取,得到了更加准确的特征表示。

表 5　消融实验评价指标

Table 5　Evaluation metrics in ablation experiments

模型	MAE	RMSE
iTransformer	3.133 9	4.332 6
TCN-iTransformer	3.049 5	4.235 4
WTConv-iTransformer	2.992 1	4.167 5
WTT-iTransformer	**2.943 8**	**4.112 8**

3　集群弹性伸缩策略验证

3.1　弹性伸缩策略优化

针对 K8s 原生弹性伸缩策略的滞后性,本文提出了 WTT-iTransformer 模型来实现对集群资源负载的预测从而为提供更加智能的弹性伸缩打下基础。同时,由于突发性负载在日常场景的常见性,且预测式策略在处理该类问题时存在一定的限制,本文提出额外引入带有突变检测的反应式扩缩策略来计算 HPA 副本数。该弹性伸缩模块具体由 3 个部分组成:

1)监控服务。监控服务基于 Prometheus 构建,其可以通过 Node_exporter 实现节点级资源指标(CPU、内存、磁盘 I/O 等)的秒级抓取,并将这些数据存储到内置的时间序列数据库中。然后基于 PromQL 的查询语言可以快速查询集群内的资源状态。

2)预测服务。本文设计的负载预测模块主要基于 Prometheus 监控系统对集群状态的实时监测的能力。具体而言,监测对象包括 K8s 集群各项资源指标以及预先配置了混合伸缩策略的所有 Deployment 实例组的历史负载参数。采集的原始数据经过预处理后,再借助 WTT-iTransformer 预测算法对未来 5 min 的集群负载和 Pod 资源情况进行精确预估。

3)混合弹性伸缩策略。当前许多自动扩缩工作往往基于对工作负载的理想化假设而成立。例如,以 HPA 为代表的反应式扩缩通常预设资源需求和负载波动具有变化不频繁的特性;而预测式主动扩缩仅当假设工作负载变化呈现出周期性规律时而成立。在真实的生产环境中,云工作负载的时变特性显著,甚至不同场景下的工作负载展现出不同的特征。但根据相关文献[17]的研究表明,大多数工作负载是可预测的,但仍存在少部分工作负载,如突发工作负载,具有随机性难以准确预测。因此,良好的自动扩缩策略既需要包含对周期规律的平稳负载的处理,又需要能够反应负载尖峰的能力。因此本文应用的弹性伸缩策略是结合反应式扩缩与预测式扩缩的混合弹性伸缩策略,通过负载预测和修改后的 HPA 控制器来计算服务所需新副本数目并进行动态调整。

由于 WTT-iTransformer 对时序负载的预测较为准确,因此混合弹性伸缩策略中的预测式扩缩部分由该模型实现。而针对突变负载的场景,本文采用优化 HPA 控制器的方法实现反应式扩缩部分。一方面设置 HPA 的一些基本参数:如 *minReplicas* 和 *maxReplicas* 足够小和足够大来应对可能的突发流量;允许 30 s 内翻倍扩容;设置 CPU 利用率阈值为 70%来触发更早扩容。另一方面通过额外实现 HPA 的突变检测算法来实现对"突变"的识别。

混合弹性伸缩策略流程如下:

在系统集群的监测过程中,系统从监控服务中提取资源对象在时间窗口 t 内的历史负载数据以及当前负载指标 *curLoad* 如内存、磁盘和当前 CPU 利用率等。接着会对得到的数据进行处理,即系统会采用预测模型计算目标指标的预测值均值 *avgPredictLoad*,本文预测的指标为 CPU 利用率。基于上述数据,系统分别计算当前反应式和主动预测式伸缩所需的新 Pod 副本数 *RR* 和 *PR*,计算公式如式(10)、(11)所示。

$$RR = curReplicas \times \left(1 + \frac{curLoad_t - curLoad_{t-1}}{curLoad_{t-1}}\right)$$
$$\Delta = curLoad_t - curLoad_{t-1} \tag{10}$$

$$PR = \left\lceil curReplicas \times \frac{avgPredictLoad}{targetMetricValue} \right\rceil \tag{11}$$

式中:*curReplicas* 是当前资源的副本数,*targetMetricValue* 是伸缩器设定指标的目标阈值,是伸缩器扩缩容的阈值。

关于突变检测算法,本文定义当变化率 Δ 超过配置值(默认 30%)时,会被识别为"突变",然后 HPA 控制器会计

算出 *RR* 供后续决策使用。

考虑到响应式和主动式伸缩策略可能产生不同决策的矛盾情况(如一种策略建议扩增副本数而另一种则建议缩减),实验采用"扩容优先"原则以解决这一问题。按照该原则,在扩增决策中选择较高的副本数作为扩容依据;而在缩减决策中,则选取较大的副本数作为最终决策,以保证系统的可用性。这一机制确保了在矛盾决策出现时的处理策略更加稳健可靠。具体而言,若资源对象的新副本数量为 *newReplicas*,当前的资源对象副本数量为 *curReplicas*,那么当 *RR* 和 *PR* 都小于 *curReplicas* 时,*newReplicas* 的计算公式如式(12)所示;当 *RR* 和 *PR* 至少有一个大于 *curReplicas* 时,*newReplicas* 的计算公式如式(13)所示。

$$newReplicas = \max(RR, PR, minReplicas) \tag{12}$$

$$newReplicas = \min(\max(RR, PR), maxReplicas) \tag{13}$$

式中:*minReplicas* 和 *maxReplicas* 是预先设定好的资源对象可以弹性伸缩的最小和最大的副本数量。

最后,弹性伸缩模块会根据计算得到的新副本数目 *newReplicas* 来调整资源对象的副本数量来实现 Pod 的扩缩容。

综上所述,通过混合弹性伸缩策略的设计可以看出该策略应明显优于传统 HPA 策略,并且其在平衡响应速度和资源利用率方面也应提升显著。首先在响应速度上,通过引入了预测模型和监控系统 Prometheus,可以实现基于历史数据和趋势的负载预测,减少对可预测的突发尖峰的响应时间,有效优化了资源利用率;同时,修改后的 HPA 控制器通过一些基本参数的修改,有效地提升了 HPA 检测指标变化和 Pod 副本修改的时间间隔,从而增强了对集群负载变化的敏感性;最后在 HPA 控制器中引入突变检测算法,能够进一步对具有随机性的突发尖峰进行检测,增强集群对随机性流量突变的调整能力。

3.2 弹性伸缩策略验证

本文弹性伸缩策略的验证基于两个相同资源的 K8s 集群进行。集群 1 使用内置的弹性伸缩算法,集群 2 使用本文提出的基于 WTT-iTransformer 的预测算法的弹性伸缩策略。每个集群的实验环境配置如表 6 所示。集群中的实验 Pod 选择 Nginx,每个 Pod 的 CPU 配额设置为 256 毫核,并设定伸缩阈值为 50%。考虑到实际生产中的负载波动的复杂性,本文采用工具 Jmeter 来进行大量流量的生成。

1)短连接应用场景

本次测试共持续 20 min,采用高负载与低负载两种模式循环两次的测试方案,每次模拟 4 000 个虚拟用户对服务进行请求。高负载测试期和低负载测试期各持续 5 min,依次交替进行。在整个实验过程中,系统每隔 10 s 对两个策略下的资源状态进行采样,具体包括系统默认

表 6 集群配置

Table 6 Cluster configuration

节点	操作系统	处理器	内存/G	磁盘大小/G
Master	CentOS7	4 Cores	2	80
Node1	CentOS7	4 Cores	2	80
Node2	CentOS7	4 Cores	2	80
Node3	CentOS7	4 Cores	2	80

HPA 策略和本文混合弹性伸缩策略下的 Pod 副本数。在高负载期间,通过 Jmeter 工具对服务进行性能测试,实时记录各采样点的响应时间。而在低负载期间,则持续向 Nginx 应用服务发送测试请求,最终获得 120 组 Pod 数量变化和服务响应时间的对比数据。图 9 展示了两种策略下 Pod 副本数的变化情况,图 10 对比了两种策略下的响应时间。

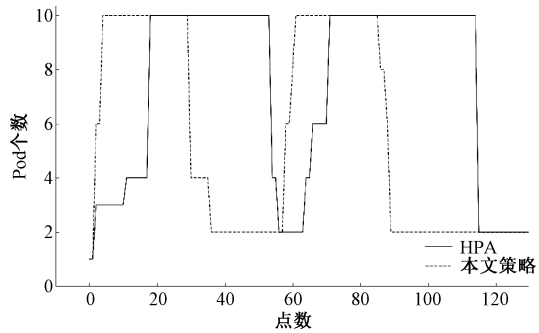

图 9 Pod 数量变化

Fig. 9 Pod replica count variation chart

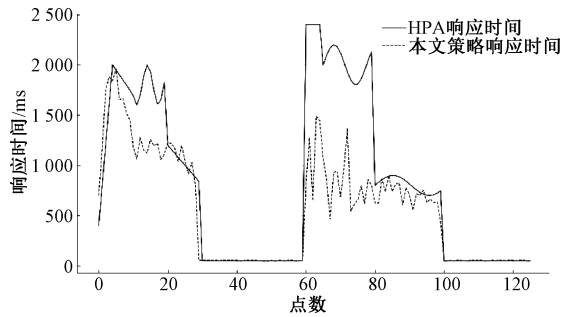

图 10 响应时间变化

Fig. 10 Response time variation chart

观察图 9、10 中数据可以发现:每 60 个数据点为一个完整的测试周期,在每个周期的前半段(即高负载阶段),本文所提出的弹性伸缩策略能够更快速地扩大 Pod 数量来实现资源扩展;而在后半段(即低负载阶段),该策略则能够更及时地进行资源收缩,从而达到优化资源配置的目的。

在第 1 个测试周期开始时,两个集群都处于初始状态,两种策略均以 1 个 Pod 实例提供服务。从响应时间的数据对比可以看出,相比于默认的 HPA 策略,本文提出的

策略能够提前约 10 个数据点完成响应时间的回落，从 1 684 ms 降至 1 115 ms 有约 33.8%的提升。而在第 1 个周期的高负载结束时，Pod 副本数快速缩减至 2 个，这一缩减动作较 HPA 策略提前了大约 25 个数据点，且在这 25 个数据点期间，响应时间相较于 HPA 策略没有产生明显增加。

进入第 2 个测试周期后，本文提出的策略表现出更快的资源扩展能力。在周期启动阶段，Pod 数量迅速从 2 个增加到 10 个，较默认策略提前了近 9 个数据点。在这段时间内，本文策略的平均响应时间 992 ms 相较于默认 HPA 策略的 2 187 ms 有 54.6%的提升，表现出更优的响应性能。

2）长连接应用场景

为进一步验证本文提出的混合弹性伸缩策略在不同类型负载下的适应性，本文选择 WebSocket 长连接服务作为实验对象。长连接实验在 K8s 集群中构建了初始 Pod 规模为 1、上限为 10 的 WebSocket 服务。相较于短连接应用，长连接应用会长期占用资源，因此在扩缩容的速度上相较于短连接的延迟会略微增加。

为模拟生产环境中的长连接应用场景，本实验采用了 WebSocket 测试工具 WebSocketKing 来生成长连接请求。实验测试时长为 20 min，每隔 10 min 建立 1 000 个长连接请求，并在 5 min 后自动断开。每个长连接设置每隔 5 s 发送数据来模拟高频数据传输，同时每隔 10 s 记录系统在不同弹性伸缩策略下的 Pod 数量变化及服务响应时间来验证本文混合弹性伸缩策略在长连接场景下的稳定性。

在长连接场景下，图 11 展示了两种策略下 Pod 副本数的变化情况，图 12 对比了两种策略下的响应时间。

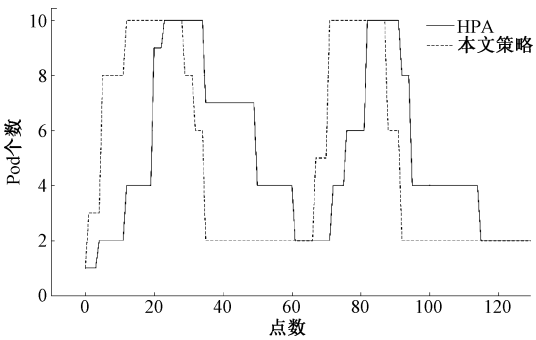

图 11　Pod 数量变化

Fig. 11　Pod replica count variation chart

观察图 11 中 Pod 数量的变化规律可以发现：在长连接的场景下，本文提出的混合弹性伸缩策略能够更迅速地根据连接请求的规模及数据传输频率来实现资源的动态调整。在高负载初期阶段，本文策略的 Pod 数增长至最大时比默认策略快约 15 个数据点即约 2.5 min，有效确保了响应时间的稳定性。同时相较于默认的 HPA 策略，本文策略在高负载阶段能快速地将 pod 数量从 1 扩容至 10，且

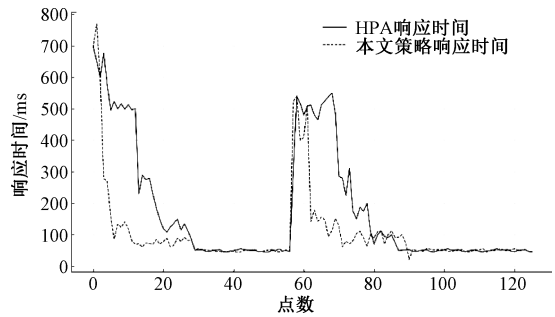

图 12　响应时间变化

Fig. 12　Response time variation chart

每个中间阶段停留时间较少；而 HPA 策略下的 Pod 副本数在初期增长较为迟缓，导致响应时间出现显著波动。

从图 12 响应时间对比图中的数据来看，在去除前 5 个非平稳数据点后，HPA 策略下的服务响应时间在负载高峰前期持续维持在 500 ms 左右，而本文提出的弹性伸缩策略将响应时间稳定在 100～200 ms。这一结果表明本文策略在处理长连接请求的高负载时更具效率，降低了请求延迟，并在负载下降阶段能够及时回收资源，实现了资源节省与系统稳定性提升的双重优化。

从图 11、12 的实验结果显示，本文提出的混合弹性伸缩策略在长连接场景下表现出更强的负载适应能力和稳定性，不仅有效缩短了响应时间，还提升了系统资源的利用效率，也进一步验证了该策略在多样化应用场景中的普适性。

综上所述，本文提出的改进混合弹性伸缩策略在长、短连接的场景下，在面对高负载时均能快速扩容来提高系统性能与服务响应速度；在高负载结束后均能确保服务质量的同时迅速缩容，实现资源的高效利用。相较于传统的 HPA 策略，展现出优异的稳定性和普适性表现。

4　结　　论

针对 K8s 集群默认弹性伸缩机制 HPA 存在的响应延迟和资源使用效能不足的问题，本文提出了一种整合预测模型算法与实时负载监控的混合弹性伸缩优化策略。具体而言，本文首先引入了创新提出的 WTT-iTransformer 模型对集群及 Pod 负载资源预测，从而有效缓解传统机制的响应延迟问题。然后在此基础上，设计并验证了基于预测结果和实时负载数据的混合弹性伸缩策略。该策略显著提高了系统的响应速度和资源利用效率，同时兼顾了负载变化的提前感知与突发性负载的处理能力。实验数据表明，所提出的 WTT-iTransformer 预测模型在资源使用趋势预测方面具有较高的准确性；与现有方案相比，提出的优化策略在负载变化的预见性和突发负载应对能力方面均表现出显著优势，体现出了更好的系统稳定性。特别是在面对负载突变时，该策略展现出了更强的容错能力和快速恢复能力。

尽管本文提出的模型已具有较好的实用性,但本文的策略仅针对单一应用场景进行了实验验证,未来研究将进一步根据集群内不同异构任务的敏感指标,通过自动化流程结合调度机制实现完全自动化的弹性伸缩和任务启停,从而进一步提升系统性能并降低能耗。

参考文献

[1] 李俊俊,董建刚,李坤. 基于 Kubernetes 的集群节能策略研究[J]. 计算机工程,2024,50(9):82-91.
LI J J, DONG J G, LI K, et al. Research on energy-saving strategies for clusters based on Kubernetes[J]. Computer Engineering, 2024, 50(9): 82-91.

[2] AL-SHARIF Z A, JARARWEH Y, AL-DAHOUD A, et al. ACCRS: Autonomic based cloud computing resource scaling[J]. Cluster Computing, 2017, 20: 2479-2488.

[3] 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 Y, HUANG J X. Elastic scaling strategy for Kubernetes-based applications[J]. Computer Systems & Applications, 2019, 28(10): 213-218.

[4] 单朋荣,杨美红,赵志刚,等. 基于 Kubernetes 云平台的弹性伸缩方案设计与实现[J]. 计算机工程,2021,47(1):312-320.
SHAN P R, YANG M H, ZHAO ZH G, et al. Design and implementation of elastic scaling scheme based on Kubernetes cloud platform[J]. Computer Engineering, 2021, 47(1): 312-320.

[5] 石硕,魏振辉,刘晓菲,等. 一种基于 LSTM 的 Kubernetes 容器云弹性伸缩策略研究[J]. 制造业自动化,2023,45(9):189-196.
SHI SH, WEI ZH H, LIU X F, et al. Research on elastic scaling strategy of Kubernetes container cloud based on LSTM[J]. Manufacturing Automation, 2023, 45(9): 189-196.

[6] 陈烨. 面向 Web 应用的 Kubernetes 容器弹性伸缩方法的研究[D]. 上海:华东师范大学,2022.
CHEN Y. Research on elastic scaling method of Kubernetes containers for web applications[D]. Shanghai: East China Normal University, 2022.

[7] FARAHNAKIAN F, LILJEBERG P, PLOSILA J. LiRCUP: Linear regression based CPU usage prediction algorithm for live migration of virtual machines in data centers[C]. 2013 39th Euromicro Conference on Software Engineering and Advanced Applications, Santander, Spain, 2013: 357-364.

[8] CALHEIROS R N, MASOUMI E, RANJAN R, et al. Workload prediction using ARIMA model and its impact on cloud applications' QoS[J]. IEEE Transactions on Cloud Computing, 2015, 3(4): 449-458.

[9] JANARDHANAN D, BARRETT E. CPU workload forecasting of machines in data centers using LSTM recurrent neural networks and ARIMA models[C]. 2017 12th International Conference for Internet Technology and Secured Transactions (ICITST), Cambridge, UK, 2017: 55-60.

[10] 覃盛. 基于 Kubernetes 的自适应伸缩的优化与实现[D]. 武汉:华中科技大学,2021.
QIN SH. Optimization and implementation of adaptive scaling based on Kubernetes[D]. Wuhan: Huazhong University of Science and Technology, 2021.

[11] BI J, YUAN H T, ZHOU M CH. Temporal prediction of multiapplication consolidated workloads in distributed clouds[J]. IEEE Transactions on Automation Science and Engineering, 2019, 16(4): 1763-1773.

[12] 朱彦民,李忠虎,王金明,等. 基于 Transformer 的风电机组故障预测[J]. 电子测量技术,2024,47(13):45-52.
ZHU Y M, LI ZH H, WANG J M, et al. Fault prediction of wind turbines based on Transformer[J]. Electronic Measurement Technology, 2024, 47(13): 45-52.

[13] 李欣宇,孙传猛,魏宇,等. 融合 Transformer 与残差通道注意力的恶劣场景水位智能检测方法[J]. 电子测量与仪器学报,2023,37(1):59-69.
LI X Y, SUN CH M, WEI Y, et al. Intelligent water level detection method for harsh scenes combining Transformer and residual channel attention[J]. Journal of Electronic Measurement and Instrumentation, 2023, 37(1): 59-69.

[14] 朱焕宇,王明泉,贾虎,等. 基于多维度动态衰减 Transformer 的轮胎检测算法应用[J]. 电子测量技术,2024,47(7):88-94.
ZHU H Y, WANG M Q, JIA H, et al. Application of tire detection algorithm based on multi-dimensional dynamic decay Transformer[J]. Electronic Measurement Technology, 2024, 47(7): 88-94.

[15] 黄星华,吴天舒,杨龙玉,等. 一种面向旋转机械的基于 Transformer 特征提取的域自适应故障诊断[J]. 仪器仪表学报,2022,43(11):210-218.
HUANG X H, WU T SH, YANG L Y, et al. A Transformer-based feature extraction method for domain adaptive fault diagnosis of rotating machinery[J]. Chinese Journal of Scientific Instrument, 2022, 43(11): 210-218.

[16] 赵昱坡，黄伟，张剑飞. 基于混合尺度健康因子的 LSTM-Transformer 锂电池寿命预测[J]. 电子测量技术，2024，47(11)：112-122.
ZHAO Y P，HUANG W，ZHANG J F. Lithium battery life prediction based on hybrid-scale health factor and LSTM-Transformer [J]. Electronic Measurement Technology，2024，47(11)：112-122.
[17] 孟春阳. 面向云原生微服务的自动扩缩算法研究[D]. 广州：中山大学，2024.
MENG CH Y. Research on auto-scaling algorithms for cloud-native microservices[D]. Guangzhou：Sun Yat-sen University，2024.

作者简介

陈奇超，硕士研究生，主要研究方向为容器技术、网络虚拟化。
E-mail：crota_chen@163.com

叶楠，教授，博士，主要研究方向为先进通信及智慧感知光子芯片及集成技术。
E-mail：aslanye@shu.edu.cn

曹炳尧(通信作者)，高级实验师，博士，主要研究方向为高速网络数据分析与处理及网络测试模拟技术。
E-mail：caobingyao@shu.edu.cn