

文章编号: 2095-2163(2022)09-0172-07

中图分类号: TP311

文献标志码: A

# 面向 Web 应用的智能监控系统的设计与实现

班晏子婧, 李 晖, 陈 梅, 孙 武

(贵州大学 计算机科学与技术学院, 贵阳 550025)

**摘要:** 目前,性能监控和数据分析对 Web 应用至关重要,但是常见的监控系统只是展示其性能指标数据,运维人员人为分析复杂海量的性能数据会增加相应的工作量。针对以上问题设计并实现了一种智能监控系统,该系统不仅能够采集并展示 Web 应用中相关资源的性能数据,而且可以根据历史数据进行趋势预测,以便运维人员提前预估故障。首先,通过压力测试工具对被测的 Web 应用产生运行负载;然后,为采集 Web 应用的资源数据部署了由 Telegraf 和 InfluxDB 构建的性能监控系统;最后,鉴于数据具有时序性的特点,系统调用了 LSTM 神经网络模型对性能数据的短期趋势进行预测,为 Web 应用性能运维提供有效、准确、及时的决策支持。

**关键词:** Web 应用; 性能监控; 时间序列预测; LSTM; 数据分析

## Design and implementation of intelligent monitoring system for Web application

BAN Yanzijing, LI Hui, CHEN Mei, SUN Wu

(College of Computer Science and Technology, Guizhou University, Guiyang 550025, China)

**[Abstract]** At present, performance monitoring and data analysis are crucial for Web application, but common monitoring systems only display their performance data, and the manual analysis of complex and massive performance data conducted by operation and maintenance personnel could increase the corresponding workload. In response to the above problems, an intelligent monitoring and prediction system is designed and implemented, which can not only collect and display performance data of related resources in Web application, but also make trend prediction based on the historical data, so that the operating personnel can predict the failure in advance. First, the operating load is generated on the tested Web application through the stress testing tool; then, a performance monitoring system built by Telegraf and InfluxDB is deployed to collect resource data of Web applications; finally, given the time series characteristics of performance data, the system calls the LSTM neural network model to predict the short-term trend of the performance data, and provides effective, accurate and timely decision support for the operation and maintenance of the Web application.

**[Key words]** Web application; performance monitoring; time series forecasting; LSTM; data analysis

## 0 引言

近年来,随着信息技术的快速发展和大数据的蓬勃兴起,人们对 Web 应用和网络平台的性能要求越来越高,倘若应用的响应时间超出用户所能接受的范围或者系统存在崩溃的风险,可能会造成用户放弃使用该 Web 应用平台的负面后果<sup>[1]</sup>,甚至导致项目彻底失败。根据亚马逊的报告,系统延迟每增加 100 ms,会导致销售额下降 1%<sup>[2]</sup>。搜索引擎、电子商务等 Web 应用服务出现故障问题会造成一定的经济损失,还会影响人们的日常生活,所以系统性能问题逐渐受到学界与技术界的关注和重视。如何高效、持续不断地监控和分析应用系统的运行状态,

现今已经成为亟待解决的重要问题之一<sup>[3]</sup>。

Web 应用的性能好坏评价需要借助可量化的数据指标来反映,基于细粒度的评价指标数据更能准确反映具体的性能问题。所以,用于采集 Web 应用平台性能指标数据的监控工具显得至关重要。目前市面上常见的监控产品有企业级分布式监控系统 Zabbix<sup>[4]</sup>、阿里巴巴的 Java 诊断工具 Arthas、监视系统运行状态和网络信息的监控系统 Nagios<sup>[5]</sup> 和 Apache 开源的应用性能管理平台 SkyWalking<sup>[6]</sup> 等。然而,大多数的性能监控平台主要是对 Web 应用平台进行性能数据的采集和可视化,并未能深入地挖掘和分析应用中性能指标的潜在价值和信息。同时,这些复杂又海量的监控数据和阈值设置的限制

**基金项目:** 国家自然科学基金(62162010, 62162011); 贵州省科技支撑计划项目(黔科合支撑[2021]一般449)。

**作者简介:** 班晏子婧(1997-),女,硕士研究生,主要研究方向:应用性能分析;李 晖(1982-),男,教授,博士生导师,主要研究方向:大规模数据管理与分析、人工智能技术及应用、高性能数据库;陈 梅(1964-),女,教授,硕士生导师,主要研究方向:大规模数据管理与分析、医学信息管理与分析服务;孙 武(1996-),男,博士研究生,主要研究方向:智能运维。

收稿日期: 2022-03-04

性都增加了运维人员的压力和工作量,当前监控系统常用的预警方式是通过设置固定阈值来实现,即当监控数据超过设定的阈值时,就会触发报警。但运维人员通过监控系统无法提前预测数据的未来趋势并规划资源,这样就很难在短时间内解决相应的问题。因此,对 Web 应用平台所使用的资源进行趋势预测即已成为热点研究内容,不仅可以深入分析性能数据中潜在的有用信息,还能帮助运维人员提前预估故障,及时采取措施,提高运维效率。

对性能指标数据进行预测是根据 Web 应用在过去一段时间内的数据表现预测未来一段时间内可能的发展趋势。监控系统中要处理的数据往往是具有时间特征的时间序列数据,而时间序列趋势预测就是先分析已有时间序列,根据发展方向和过程找到时间趋势规律并得出一定模式,最后用模式来预测将来的情况<sup>[7]</sup>,例如预测网站系统的用户增长趋势和 CPU 使用率等。常见的时间序列趋势预测算法有神经网络算法、ARIMA 自回归移动平均模型、贝叶斯模型等。Sezer 等人<sup>[8]</sup>在 2018 年基于二维卷积神经网络提出 CNN-TA 算法用于金融交易的趋势预测。Lu 等人<sup>[9]</sup>在 2020 年基于 ARIMA 时间序列算法预测锂离子电池 RUL 趋势。Sara 等人<sup>[10]</sup>在 2021 年采用贝叶斯模型预测儿童 Covid-19 严重程度趋势。而循环神经网络 RNN 是一种专门用于处理序列数据的神经网络,1997 年 Hochreiter 等人就在 RNN 的基础上提出了增加记忆单元的长短期记忆神经网络 LSTM,可以控制历史信息的记忆程度,相比普通的 RNN 能够在更长的序列中有更好的表现。Song 等人<sup>[11]</sup>在 2018 年使用 LSTM 网络模型对服务器平均 CPU 利用率进行多步预测,并在真实数据集中进行验证。谭宇宁等人<sup>[12]</sup>在 2019 年设计了一种基于 LSTM 神经网络的软件老化资源预测方法,对 Web 服务器资源消耗进行预测。赵春生等人<sup>[13]</sup>在 2021 年利用 LSTM 算法对服务器内存使用率、磁盘使用率和消息队列数等性能指标进行预测,实验证明预测值与真实值之间的均方根误差较小。从中可以发现,对于具有时序性特点的性能数据,LSTM 神经网络在预测精度和稳定性方面上都有一定的优势。

因此,为保障 Web 应用的稳定运行,降低企业成本,提高运维人员的工作效率,本文设计并实现了一种面向 Web 应用的智能监控系统。研发后的系统具有采集应用性能指标数据以及对性能数据进行基于 LSTM 模型趋势预测的功能。

因此,为保障 Web 应用的稳定运行,降低企业成本,提高运维人员的工作效率,本文设计并实现了一种面向 Web 应用的智能监控系统。研发后的系统具有采集应用性能指标数据以及对性能数据进行基于 LSTM 模型趋势预测的功能。

### 1 关键技术工具

由 Telegraf 和 InfluxDB 构建的运维监控平台作为可快速搭建的轻量级实时监控系统,目前已广泛用于各类市场<sup>[14]</sup>,主要针对运维监控数据量大、实时、高效的存储需求而提出的。Telegraf 和 InfluxDB 运维监控平台架构图如图 1 所示。本文选用 Telegraf 和 InfluxDB 工具实现智能监控系统的监控功能。对此拟做阐释论述如下。

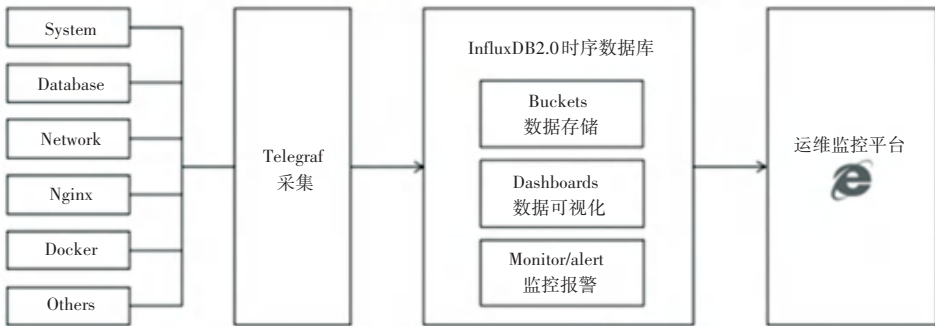


图 1 Telegraf 和 InfluxDB 运维监控平台

Fig. 1 Telegraf and InfluxDB O&M monitoring platform

Telegraf 是一个开源的数据采集工具,是 InfluxData 公司的时间序列平台 TICK 技术栈中的一部分,使用 Go 语言实现,主要用于从运行的系统中采集时间序列型数据指标,其中包括 CPU、内存、磁盘和网络带宽等相关资源的性能数据,还可以根据需求生成相应的配置文件。同时,Telegraf 提供了广

泛的插件,会定时执行输入插件以采集数据,数据经过处理插件和聚合插件进行清理和聚合,继而批量转发到各种其他数据存储、服务和消息队列,包括 Kafka、InfluxDB、MQTT、OpenTSDB 等。Telegraf 不需要自己手工编写脚本,定时采集且配置简单,可快速完成监控数据采集,并获取到多种组件的运行信息。

InfluxDB 是一个开源的时间序列平台,其核心是支持与时间有关的求和、最小和最大等一系列函数的分布式时序、事件和指标数据库,专门用于处理由应用程序和基础设施生成的海量数据和无数来源的时间戳数据。InfluxDB 是由 Go 语言编写,不需要外部依赖,设计目标是为了实现分布式和水平伸缩扩展,内置 HTTP API,方便存储和检索,允许非常灵活的数据查询<sup>[15]</sup>。并具有可度量性和可扩展性,可以实时对大量数据进行计算,同时支持任意的事件数据。

Telegraf 和 InfluxDB 构建的轻量级性能监控平台通过实时数据采集和可视化面板提高了运维效率和水平,不仅实现了运维工作便捷化管理,同时也能够适当降低企业成本。因此,本文采用这种轻量级运维监控平台作为智能监控系统的监控部分,先使用 Telegraf 代理器采集被测系统的性能数据,再利用 InfluxDB 数据库针对采集得到的时间序列数据完成分析计算和存储等操作。

## 2 相关理论概述

时间序列数据是一串按时间维度索引的数据,用于描述某个或者某些随机变量随时间发展的变化趋势,通常时间间隔为 1 s、1 min 等恒定值。时间序列数据实际应用场景丰富,广泛用于地震预测、通信工程、数理统计及绝大多数涉及到时间数据测量的应用科学与工程学中。而时间序列预测问题的核心就是从这些数据中挖掘出其历史规律,并用其对未来的数据做出估计<sup>[16]</sup>。时间序列分析主要是根据获取的历史数据变化趋势对未来发展进行预测,而基于对这种数据的分析方法,建立 Web 应用性能数据的预测模型,预测性能数据在未来一段时间内的发展情况,以便辅助应用的运维人员进行决策,同时提供数据依据。

神经网络是目前对于时间序列数据而言常用的预测方法之一。神经网络算法是由多个神经元交叉成多个网络节点、节点层相互连接构建成的一个多维度复杂网络虚拟空间,可以存储大量的形式多样化的数据信息。在传统的神经网络模型中,层与层之间是全连接的,每层之间的节点是无连接的,这种模型对时间序列数据处理十分低效<sup>[17]</sup>。而循环神经网络(RNN)是一种专门用于处理序列数据的神经网络,比一般的神经网络要更能够处理序列变化的数据,RNN 网络会对前面的信息进行记忆,保存在网络的内部状态中,并应用于当前输出的计算中,这种记忆的能力大大增强了神经网络对时间序列问

题的学习和泛化输入能力。

本文采用的预测方法是长短记忆神经网络 LSTM,这是一种特殊的循环神经网络,允许数据在网络中向前和向后流动。LSTM 主要是为了解决 RNN 存在的长距离依赖问题和在长序列训练过程中会出现的梯度消失、梯度爆炸问题,相比普通的 RNN 能够在更长的序列中有更好的表现。LSTM 模型单元结构如图 2 所示。

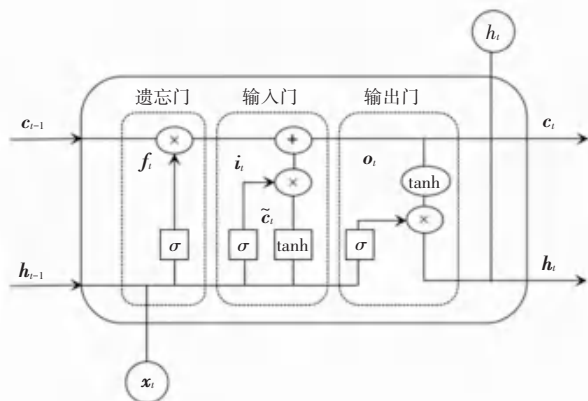


图 2 LSTM 单元结构

Fig. 2 LSTM cell architecture

LSTM 在 RNN 的基础上增加了一个细胞状态  $c_t$ , 细胞状态保存的是当前时刻隐藏状态信息,隐藏状态信息包含上一时刻的隐藏状态和当前时刻的临时隐藏状态。LSTM 通过遗忘门、输入门和输出门三个门来控制传输状态<sup>[18]</sup>。总的来说,遗忘门控制上一时刻的单元状态  $c_{t-1}$  里的信息是否保留到当前时刻细胞状态  $c_t$ , 记为  $f_t$ ; 输入门控制细胞状态中信息的加入,记为  $i_t$ ; 输出门控制当前时刻细胞状态  $c_t$  中的信息是否输出为当前隐藏状态  $h_t$ , 记为  $o_t$ 。在  $t$  时刻, LSTM 模型前向计算方法可以表示为:

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \quad (1)$$

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \quad (2)$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad (3)$$

$$\tilde{c}_t = \tanh(W_c[h_{t-1}, x_t] + b_c) \quad (4)$$

$$c_t = f_t c_{t-1} + i_t \tilde{c}_t \quad (5)$$

$$h_t = o_t \tanh(c_t) \quad (6)$$

其中,  $\sigma$  是 sigmoid 激活函数,用于决定保留多少信息进入下一记忆单元;  $\tanh$  是双曲正切激活函数,用于将任意值映射到  $[-1, 1]$  区间;  $W_*$  和  $b_*$  分别为对应门限的权重系数矩阵和偏置项。

计算过程中,矩阵在最开始时随机生成,接着在传递的过程中每次都通过反向传播不断迭代,得到准确的矩阵后对前一单元的信息进行保留和更新<sup>[19]</sup>。

由于运维监控平台采集得到的性能数据属于长时间序列,而 LSTM 模型是最适合利用长时间序列之间的关系来预测趋势。所以,本文在面向 Web 应用的智能监控系统中采用了基于 LSTM 的时间序列神经网络模型,对经过数据处理的 Web 应用平台性能数据进行时间序列预测,得到接下来一段时间性能数据变化趋势,为 Web 应用的运维工作提供参考。

## 3 系统设计

### 3.1 功能模块设计

本文系统的功能模块如图 3 所示。由图 3 可知,对系统中各组成部分模块的设计功能可逐项展开分述如下。

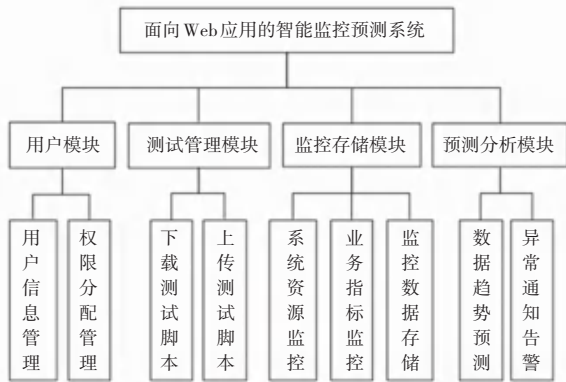


图 3 智能监控系统功能模块

Fig. 3 Intelligent monitoring system functional modules

(1)用户模块。智能监控系统主要面向 Web 应用的开发人员、测试人员以及运维人员,不仅可以对用户信息进行管理,还可以对不同的特殊用户分配不同模块的使用权限。

(2)测试管理模块。智能监控系统管理针对被测 Web 应用设计并录制的 Jmeter 测试脚本,运维人员可以根据自身实际情况和工作环境上传和下载测试脚本,以便后期进行复用。这些测试脚本用于模拟用户使用 Web 应用的真实场景,产生 Web 应用运行的工作负载。

(3)监控存储模块。智能监控系统根据 Web 应用运行的性能状况采集其资源指标数据,并存储至时间序列数据库 InfluxDB 中,前端将通过图表形式展现结果实现数据可视化以便运维人员观察,同时为预测分析模块提供数据支撑。当需要排查故障时也可以通过数据库调取历史性能数据,但由于数据不需要保存太长时间,智能监控系统将数据保留策略设置为 7 天,并定期清除过期数据。

(4)预测分析模块。运维人员可以根据监控采

集的核心性能指标选择时间范围内的数据,系统会调用 LSTM 神经网络模型进行趋势预测,每次预测使用最新 60 个时间点的值预测未来一个时间点的值,接着将最新预测的值作为已知值继续预测,直到成功预测未来 60 个时间点的值停止,并将结果通过 ECharts 折线图展示,运维人员可以根据预测的趋势提前做好预防措施。

### 3.2 系统总体架构设计

面向 Web 应用的智能监控系统总体架构分为 4 层,也就是:数据采集层、数据处理层、数据应用层和数据展示层。基本架构如图 4 所示。这里,将对各部分架构依次给出研究论述如下。

在数据采集层中,通过 Telegraf 数据采集工具采集 Web 应用平台运行状况下服务器、网络和中间件等多个维度的性能指标数据,采集过程中可以设置刷新频率、时间显示等范围,同时本文基于 JMeter 测试工具和自动化脚本,模拟用户在使用 Web 应用平台时的运行状况,从而持续产生应用运行的工作负载。在建立对 Web 应用平台的测试环境时,需要考虑真实的业务需求与用户实际使用情况,从而设计多种用户访问不同的应用场景,以使测试接近真实的运行状态,为对 Web 应用进行性能分析获取准确的数据支撑。

在数据处理层中,通过 Telegraf 数据采集工具采集 Web 应用平台运行时的细粒度性能指标数据,再利用 InfluxDB 时间序列数据库进行存储,其中指标数据主要包括硬件资源开销、容器启动时间、进程数量和平均负载数值等。将所采集到的数据通过数据清洗(如空值处理和修改字段名)、数据聚集(如统计一段时间内系统资源使用情况)以及内置函数计算等方式进行数据处理。同时,本文将 InfluxDB 时序型数据库部署到智能监控系统的服务器上,以便接下来对性能指标数据进行可视化。

在数据应用层中,智能监控系统结合从应用服务器中采集的细粒度指标数据对 Web 应用平台进行综合分析,采用单层隐藏层的 LSTM 神经网络模型对性能数据进行趋势预测,为运维工作提供数据支撑。同时,只有监控数据对于运维工作来说还远远不够,大量多维的数据会增加运维人员的工作压力,因此还需要有报警模块来进一步做好预警处理和分析。在 InfluxDB 时序数据库的 Monitoring/Alerting 模块中根据不同的监控项设置了报警规则,当被监控的时间序列数据达到规则设定的阈值时,就会发布告警信息。在进行趋势预测分析时,如果

Web 应用的服务器可能出现异常情况时,智能监控系统会通过弹框方式对运维人员进行提醒,真正让整个运维监控工作实现了智能化。

在数据展示层中,智能监控系统先通过 InfluxDB 数据库中提供的 Dashboard 编辑工具展示 Telegraf 采集的监控数据,通过图表形式将综合分析的性能数据和可能存在的异常数据(如硬件资源居高不下)展示到系统前端。同时,对于趋势预测分析后的结果,系统利用 ECharts 可视化工具将最新 60 个时间点和预测后 60 个时间点的数据以折线图方式报告给运维人员。通过数据可视化,运维人员能够快速了解到 Web 应用平台整体的性能状况,为运维人员后期排查性能瓶颈和决策性能问题提供了相应的数据支持,减少运维时间,提高工作效率,还能协助开发人员对 Web 应用平台进行更新优化,方便观察性能变化趋势。

Vue.js 框架和 Element UI 搭建,前后端使用的集成开发环境分别为 WebStorm 和 IntelliJ IDEA。

### 4.2 性能数据监控采集

本文搭建了某 Web 应用作为测试系统,该应用基于 Java 开发,可直接在应用页面进行数据集成、可视化分析和数据挖掘等数据分析操作,并新增了 500 个用户专门用于测试。首先使用压力测试工具 Jmeter 录制并运行相应的操作脚本,模拟 500 个用户在应用平台上进行实验操作,以产生应用负载。其次部署了 Telegraf 数据采集工具和 InfluxDB 时序数据库作为智能监控系统的监控模块,利用采集工具获取 Web 应用运行时的性能数据,同时根据本文系统的需求修改 Telegraf 中相应配置文件,比如将写入间隔设置为 10 s。Telegraf 代理器启动后会定时执行输入插件收集各种时间序列型数据,接着数据经过处理插件和聚合插件,这里的处理插件是对收集到的数据流进行简单的处理,例如给所有指标去重、重命名、格式转换等。聚合插件是对一段时间内流经该插件的指标数据流进行处理和平均值计算。最后,经过输出插件将处理后的批量数据输出到数据存储系统 InfluxDB 中。

InfluxDB 的数据模型与关系型数据库不同,Measurement 是存储数据的容器,包含了时间戳列 Timestamp、维度列 Tags 和数值列 Fields,类似于关系型数据库中表的概念。图 5 为 InfluxDB 数据模型的示例。分析图 5 可知,该数据集记录了 Web 应用服务器 CPU 的使用情况。其中,容器 Measurement 的名称为“cpu”,维度列 Tags 由服务器的 CPU 编号,即 cpu 列和主机名 host 列组成,数值列 Fields 由 IO 等待使用 CPU 占比 time\_iowait 列、用户使用 CPU 占比 time\_system 列、系统使用 CPU 占比 time\_user 列等构成,其内容为各项指标的数据值,time 为时间戳列。

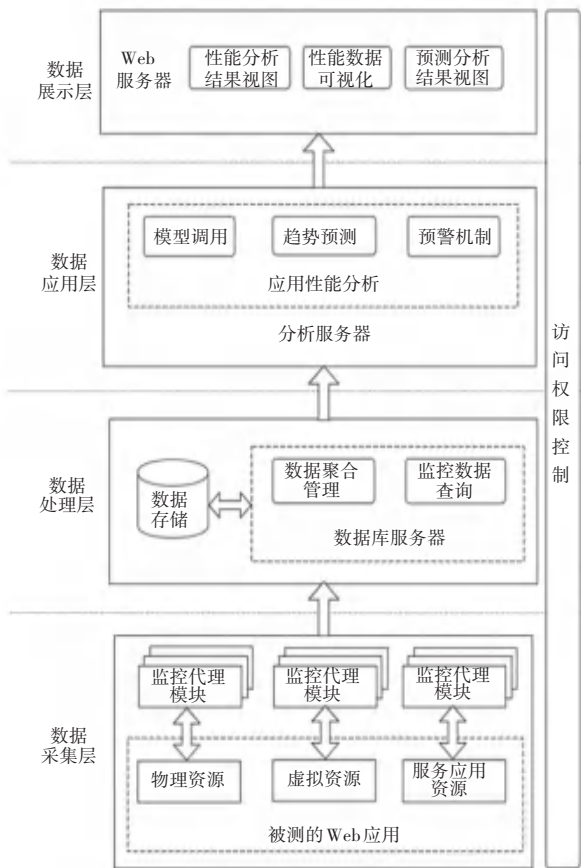


图 4 智能监控系统架构

Fig. 4 Intelligent monitoring system architecture

## 4 系统实现

### 4.1 开发环境介绍

面向 Web 应用的智能监控系统后台开发构建选择的是 Java 语言和 Spring boot 框架,前端使用

time	cpu	host	time_iowait	time_system	time_user
1558548310000000000	cpu-total	localhost.localdomain	47.2	42.12	71.9
1558548320000000000	cpu0	localhost.localdomain	18.58	47.48	46.84
1558548330000000000	cpu1	localhost.localdomain	18.56	46.7	46.54
1558548340000000000	cpu-total	localhost.localdomain	47.2	42.19	71.82
1558548350000000000	cpu0	localhost.localdomain	18.58	47.47	46.58
1558548360000000000	cpu1	localhost.localdomain	18.56	46.73	46.58
1558548370000000000	cpu-total	localhost.localdomain	47.21	42.2	71.84
1558548380000000000	cpu0	localhost.localdomain	18.54	47.48	46.57
1558548390000000000	cpu1	localhost.localdomain	18.56	46.72	46.57
1558548400000000000	cpu-total	localhost.localdomain	47.21	42.21	71.84
1558548410000000000	cpu0	localhost.localdomain	18.54	47.48	46.57
1558548420000000000	cpu1	localhost.localdomain	18.57	46.74	46.58

图 5 InfluxDB 数据模型

Fig. 5 FluxDB data model

采集的细粒度性能指标数据维度为 86,其中包括 CPU 类指标 7 个,比如用户用量百分比、系统用

量百分比、软中断时间占比等;内存类指标 20 个,比如活跃使用的内存总数、已用内存数、文件 buffer 内存数等;磁盘类指标 41 个,比如磁盘用量、使用的 inode 数量、运行中的每秒 IO 数据量、磁盘读取总用时等;系统负载类指标 3 个,比如 1 min 平均 load

值、5 min 平均 load 值等;进程类指标 4 个,比如回收中的进程数、暂停状态进程数等;以及网络类指标 11 个,比如网卡收丢包数量、网卡发包错误数量等。以性能数据中 CPU 类指标为例的数据见表 1。

表 1 CPU 类型性能指标数据信息

Tab. 1 CPU type performance indicators data information

监测时间	用户使用 CPU 百分比	系统使用 CPU 百分比	虚拟进程竞争等待使用 CPU 百分比	软中断使用 CPU 百分比	IO 等待使用 CPU 百分比	优先级调度程序使用 CPU 百分比	空闲 CPU 百分比
2021/10/8 14:00:00	64.065 4	6.607 9	0.062 9	0.125 8	0.062 9	0	29.075 1
2021/10/8 14:00:10	60.716 4	10.834 9	0.063 7	0.191 2	2.039 5	0.382 4	25.771 9
2021/10/8 14:00:20	70.643 9	6.881 3	0.063 1	0.189 3	0.063 1	0	22.159 3
2021/10/8 14:00:30	66.313 3	6.119 1	0.044 1	0.201 8	0.044 1	0	27.277 6
2021/10/8 14:00:40	65.576 6	9.125 4	0.018 9	0.075 7	0.341 0	0.145 2	24.717 2
2021/10/8 14:00:50	52.875 1	14.746 2	0.064 7	3.950 7	0.064 7	0	28.298 6

### 4.3 性能数据趋势预测

鉴于采集到智能监控系统数据库中的性能指标数据具有时序性的特点,本文通过使用 LSTM 神经网络模型对其未来发展趋势进行时间序列预测。在系统实现过程中,使用 Tensorflow 的 Java 版本 API 接口实现 Java 调用 LSTM 神经网络模型,先将 Tensorflow 的 jar 包导入 Maven 项目的 POM 文件中,在项目中放入模型文件后创建 MilkPowderUtils 接口类和静态模型调用方法,并使用 SaveModelBundle 类完成 Java 程序和模型文件之间的连接。以预测过程为参数,实例化输出类,同时将创建的形参传入,且转换为 Tensor 类型,赋值给 Tensor 对象 *input\_x*。接着,调用模型会话对象和输入数据,计算预测结

果,再将其处理后作为方法返回值。Java 程序查询数据库中要预测的数据,在标准化处理和封装后,就会调用静态模型方法,传递封装的数据,得到相应的预测值,最后使用 ECharts 可视化反标准化后的预测结果。

### 4.4 性能数据可视化

图 6 是面向 Web 应用的智能监控系统中性能指标数据的可视化界面。首页将采集的核心性能指标进行展示,运维人员可以继续通过需求在侧方导航栏内选择想要查看的相应类型的资源指标性能数据,比如 CPU 类型、内存类型、磁盘类型等,系统会根据用户的选择展示该类型下的细粒度性能数据。



图 6 智能监控系统首页

Fig. 6 Homepage of the intelligent monitoring system

同时,对于需要进行趋势预测的性能指标,运维人员可以通过点击界面按钮开启预测分析,智能监控系统会调用 LSTM 模型对选择的性能指标项进行

预测,根据最新 60 个时间点的值来预测未来 60 个时间点的数值趋势,性能指标数据趋势预测结果如图 7 所示。

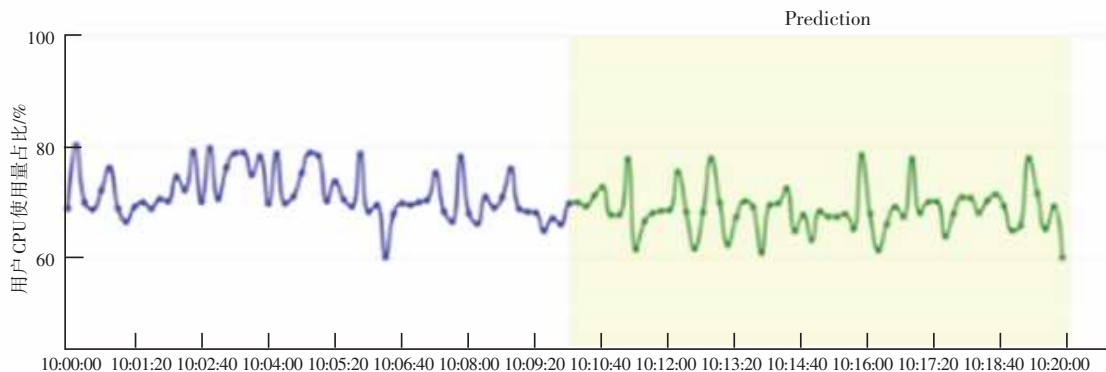


图7 智能监控系统预测结果

Fig. 7 Intelligent monitoring system prediction results

## 5 结束语

目前,在 Web 应用发展复杂的大环境下,本文设计并实现了一种不仅能够监控 Web 应用性能指标数据,同时还可以对性能数据进行基于 LSTM 模型趋势预测的智能监控系统,以便运维人员随时监测 Web 应用的整体性能情况和运行状况,在出现问题和故障时也能够通过监控系统辅助排查,而利用神经网络根据历史的性能数据预测出未来趋势,更利于提前做好准备工作和预防措施。在未来研究和开发的过程中,还需要更进一步地提高监控指标数据的精确度和预测网络模型的准确率。

## 参考文献

[1] 杨秀杰. 政务大数据服务系统性能测试技术研究及平台实现[D]. 北京:北京交通大学,2021.

[2] LAI C A, KIMBALL J, ZHU Tao, et al. MilliScope: A fine-grained monitoring framework for performance debugging of n-tier Web services[C]// 2017 IEEE 37<sup>th</sup> International Conference on Distributed Computing Systems (ICDCS). Atlanta, GA, USA: IEEE, 2017:92-102.

[3] 余强. 基于半监督学习的应用性能分类分析系统的研究与实现[D]. 北京:北京邮电大学,2018.

[4] PEIXIAN C, BIAN Shenghua, ZHENG Hongliang, et al. Research on cluster monitoring and prediction platform based on Zabbix technology[J]. IOP Conference Series Earth and Environmental Science, 2020, 512:012155.

[5] ANTONIOS A, CHATZIPANTOU N. Using nagios on a Raspberry Pi to monitor a network with emphasis on security[J]. WSEAPP Transactions on Computers, 2021, 19:262-267.

[6] 吴晟. Apache SkyWalking, 开源监控生态[J]. 软件和集成电路, 2021(6):62.

[7] 彭冬,朱伟,刘俊,等. 智能运维从0搭建大规模分布式 AIOps 系统[M]. 北京:电子工业出版社,2018.

[8] SEZER O B, OZBAYOGLU A M. Algorithmic financial trading with deep Convolutional Neural Networks: Time series to image conversion approach[J]. Applied Soft Computing, 2018, 70:525-538.

[9] LU Cun, GU Zhengjian, YUAN Yan. RUL prediction of lithium ion battery based on ARIMA time series algorithm[J]. Materials Science Forum, 2020, 999:117-128.

[10] SARA D R, VILLAVARDE S, SANZ-SANTAEUFEMIA F J. A Bayesian model to predict COVID-19 severity in children[J]. The Pediatric Infectious Disease Journal, 2021, 40(8):e287-293.

[11] SONG B, YU Y, ZHOU Y, et al. Host load prediction with long short-term memory in cloud computing[J]. The Journal of Supercomputing, 2018, 74(12):6554-6568.

[12] 谭宇宁,党伟超,白尚旺,等. LSTM 网络模型在 Web 服务器资源消耗预测中的应用研究[J]. 计算机系统应用,2019,28(07):214-220.

[13] 赵春生,梅松,张双江,等. 基于 XGBoost 和 LSTM 的智能监控系统研究[J]. 电子技术与软件工程, 2021(3):156-158.

[14] 闫俊. 基于 InfluxDB+Telegraf 的运维监控平台建设研究[J]. 市场周刊·理论版, 2020(33):180-181.

[15] 徐化岩,初彦龙. 基于 influxDB 的工业时序数据库引擎设计[J]. 计算机应用与软件,2019,36(09):33-36,40.

[16] 杜嘉薇,周颖,郭荣华,等. 网络安全态势感知提取、理解和预测[M]. 北京:机械工业出版社,2018.

[17] 贵福胜,王锋. 基于 LSTM 的 Web 应用系统性能预测[J]. 计算机科学,2019,46(10A):183-187.

[18] 党伟超,李涛,白尚旺. 基于 LSTM 网络的 Web 软件系统实时剩余寿命预测[J]. 计算机系统应用,2021,30(07):253-258.

[19] 张双江. 基于 XGBoost 和 LSTM 的智能监控系统的设计与实现[D]. 南京:南京大学,2019.