马志宇,吴　颖,夏　川,等. 基于改进 DRF 算法的农业微服务负载均衡[J].江苏农业学报,2020,36(5):1298-1304.
doi:10.3969/j.issn.1000-4440.2020.05.029

基于改进 DRF 算法的农业微服务负载均衡

马志宇[1]，吴　颖[1]，夏　川[1]，刘　飞[1,2]，吴云志[1,2]，乐　毅[1,2]，张友华[1,2]
(1.安徽农业大学信息与计算机学院,安徽 合肥 230036；2.安徽农业大学/安徽省北斗精准农业信息工程实验室,安徽 合肥 230036)

摘要： 微服务作为一种新型的互联网平台架构设计,在现代智慧农业中具有细粒度、低耦合度和高可靠性的优势。在负载均衡智慧农业微服务领域,目前针对复杂农业环境下的特定微服务负载均衡方案的研究较少,而高性能微服务负载均衡策略能够有效提升农业物联网的网络体验和性能指标。本研究提出一种改进的主导资源公平分配(Dominant resource fairness，DRF)算法,通过引入多角度的性能测评因素,着力于提升在智慧农业平台的微服务化下实现服务高效负载均衡的性能。经测试,本研究提出的算法相比于目前流行的微服务平台自带的负载均衡算法在响应时间、吞吐率和稳定性方面有明显提升,实现了各个农业生产单位物联网设备和数据终端以低耦合度的方式接入系统平台,同时实现高效地利用、治理和规范化来自各个数据源的农情数据,具有较高的实用价值。

关键词： 负载均衡；微服务；物联网；智慧农业；主导资源公平分配(DRF)算法；
中图分类号： F303.3　　**文献标识码：** A　　**文章编号：** 1000-4440(2020)05-1298-07

Agricultural microservice load balancing based on improved dominant resource fairness (DRF) algorithm

MA Zhi-yu[1]，WU Ying[1]，XIA Chuan[1]，LIU Fei[1,2]，WU Yun-zhi[1,2]，YUE Yi[1,2]，ZHANG You-hua[1,2]
(1.*School of Information & Computer*,*Anhui Agricultural University*,*Hefei* 230036,*China*；2.*Anhui Agricultural University*，*Anhui Provincial Engineering Laboratory of Beidou Precision Agriculture Information*,*Hefei* 230036,*China*)

Abstract： Microservice is a new type of architecture design for internet platform，with the advantages of fine granularity，low coupling degree and high reliability in modern smart agriculture. In the field of load balancing of smart agricultural microservices，there are few researches on load balancing schemes of specific microservice under complex agricultural environments in the current，and load balancing strategies of efficient microservice can effectively improve the network experience and performance index of agricultural internet of things (IOT). This paper proposed an improved dominant resource fairness (DRF) algorithm and focused on improving the performance of service efficient load balancing under micro-servitization of smart agriculture platform by introducing multi-angle performance evaluation factors. After testing，the algorithm proposed in this paper showed significant improvement in response time，throughput and stability compared with the load balancing algorithms provided by current popular microservice platforms. IOT equipment and data terminal of each agricultural production unit can be connected to the system platform in a mode of low coupling degree，while agricultural data from various data sources can be effectively utilized，managed and standardized. The algorithm proposed in this paper shows good practical value.

Key words： load balancing；microservices；internet of things；smart agriculture；dominant resource fairness (DRF) algorithm

收稿日期：2020-03-20
基金项目：国家重点研发计划项目(2017YFD0301303)；安徽省级大学生创新训练计划项目(201910364204)
作者简介：马志宇(1998-)，男，安徽蚌埠人，本科，研究方向为大数据技术、智慧农业、人工智能。(E-mail)mzy@ahau.edu.cn
通讯作者：乐　毅，(E-mail)yyyue@ahau.edu.cn；张友华，(E-mail)zhangyh@ahau.edu.cn

近年来,市场上的智慧农业平台大都采用单体架构模式(图 1)实现,各个业务逻辑往往互相交叉、依赖,在开发中一个模块的故障很可能造成整个系统的瘫痪[1]。

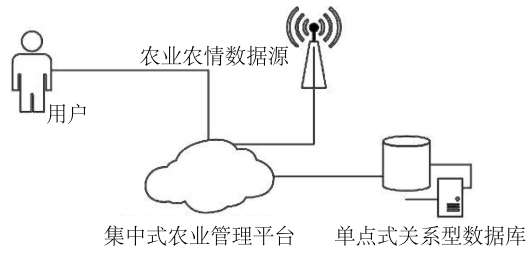

图 1 传统智慧农业平台架构

Fig.1 Traditional smart agricultural platform architecture

传统的服务端负载均衡方法是通过 Nginx 之类的工具在服务端对客户端请求进行转发从而实现的,由于所有数据均需要访问同一地点的 Nginx 服务,对于负载均衡工具本身而言,各地农场大量的并发请求会形成聚集从而形成性能的瓶颈[2]。对于分散在全国各地的用户来说,所有请求都需要首先访问单一节点的负载均衡工具,其中网络延迟和数据多次转发形成的丢包率使得这种负载均衡方式的运行效果不理想。

面对大型农业生产集团遍布各地的一线生产单元,不同地域的物联网设备因各自网络环境的不同,对单点式服务器的访问性能必然有相当大的差距。如何实现农业数据的稳定高效传送成为亟需解决的问题。微服务(Microservice)最早由 Fowler 与 Lewis[3]于 2014 年共同提出,微服务作为一种架构模式,其倡导将系统内各功能解耦到独立的服务中并在不同的进程中运行。不同服务与业务之间可以采用不同的编程语言与数据存储技术[4],大大解决了随着单一系统业务逻辑复杂度增长而带来的扩展性、稳定性问题[5]。

在具体农业生产场景中,物联网传感器数据源、农业监控摄像头在大型农业生产主体中往往是离散分布在不同农场甚至不同城市中[6],大量传感器和网关会频繁地与服务器进行实时通信[7],微服务客户端负载均衡技术则可以实现由各个用户根据自身网络情况自主判断选择对其网络环境最佳的服务器进行通信,从而优化不同地域用户的网络感知体验。

对一线用户而言,在农业生产的全过程中很难使用同一个厂商的产品,单体架构的农业系统很难做到及时与不同品牌的农业设备兼容,而微服务的设计思想同时可以实现将农业生产全程物联网设备和数据终端统一接入、共享管理[8]。

对于微服务的负载均衡而言,其算法逻辑在客户端软件中执行,客户端首先在服务注册中心订阅消息,注册中心向客户端定时推送服务列表,客户端实时根据最新的服务列表执行负载均衡算法[9]。

1 研究框架

针对目前智慧农业信息系统和微服务均衡方案存在的问题,本研究设计了一套基于负载均衡改进主导资源公平分配(DRF)算法的智慧农业开放平台。本平台以 Spring Cloud 微服务框架为基础,结合了 Eureka 服务注册中心、Ribbon 负载均衡工具,实现了一套基于客户端的高性能农业微服务负载均衡解决方案。平台架构见图 2。

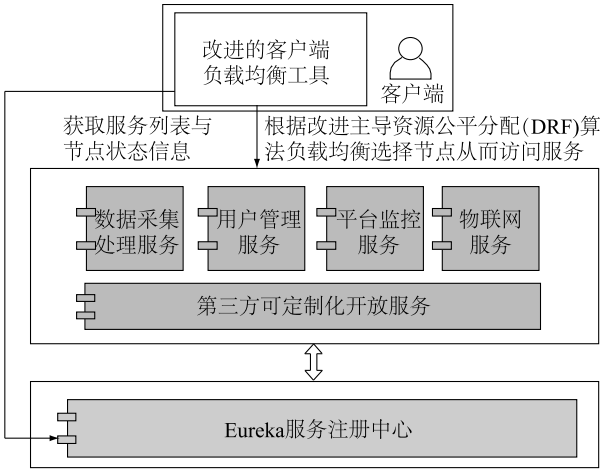

图 2 本研究改进的平台架构

Fig.2 The improved platform architecture of this study

通过构建多个微服务取代传统 MVC[模型(Model, M)-视图(View, V)-控制器(Controller, C)]设计模式的 Model 与 View 层,本研究设计了复用性高、与平台运行密切相关的数据采集处理服务、用户管理服务、平台监控服务、农业物联网服务作为系统的核心微服务。针对农、林、牧、渔不同行业的专有需求,软件开发商和开发者可以以微服务的形式独立开发后与平台直接对接。

Eureka 是一个服务发现技术,可实现在服务中的负载均衡与故障转移,每个服务在启动时会主动向服

务注册中心注册[10],服务注册中心将会接收到此服务所在服务器的主机号、端口号和通信协议等信息。当平台中的各个应用(消费者)需要调用微服务时,首先向服务注册中心订阅该微服务,同时注册中心会返回该服务的地址给消费者。与此同时,各个微服务会定期向注册中心发送心跳包以报告自己的服务状态。

Ribbon 是一个基于客户端的负载均衡工具,客户端的所有请求将通过 Ribbon 定时从服务注册中心拉取服务列表[10],经过负载均衡算法选择节点后转发给服务器,该工具自带数种负载均衡方案,但都普遍缺乏智慧农业以及农业物联网环境下的针对性。基于本研究提出的算法,通过修改其部分源码,完成了在 Ribbon 下的改进 DRF 算法。

2 基于改进 DRF 算法的微服务负载均衡算法设计

针对农业环境下不同生产单位[如分散在全国各地、使用不同网络接入方式的农场数据终端、移动个人数字助理(PDA)以及网关传感器]访问单一节点的服务可能因为不同因素影响服务的可靠性,因此负载均衡算法需要充分考虑到延迟时间、不同节点当前负载、连通率等指标,确保不同地区、网络环境和接入方式的用户能够获得最佳体验。

2.1 评价因素的建立

建立负载均衡评价因素集合(X),

令 $FailureCount$=失败请求次数,

$ActiveRequestsCount$=当前活跃的连接数,

$SuccessiveConnectionFailureCount$=连续失败的连接数,

$ResponseTimeStdDev$=响应时间标准差,

$Latency$=从客户端到主机的延迟时间,

从而得到:

$X=\{FailureCount, ActiveRequestsCount, SuccessiveConnectionFailureCount, ResponseTimeStdDev, latency\}$

2.2 评价因素的规范化处理

评价因素体系从不同方面体现了各个节点的性能特征,有的指标从正面说明总体,有的指标相反,因此需要对数据进行趋同化。对于不同性质和单位的指标不能直接计算,其不能反映不同指标因素的共同作用结果,因此还要考虑将所有指标对衡量结果的影响无量纲化,再进行相应计算才可得出正确结果[11]。为了使评价因素无量纲化与同趋化,针对 5 种不同的评价因素分别进行标准化操作。

Z-score 模型算法,也称作标准分数算法(Standard score),其计算方法为用样本中一个数与平均数的差除以标准差,能够真实地反映一个分数距离平均数的相对标准距离,其被看作一个数据点的值和总体平均测量值的标准偏差[12],转化后的每个标准分数总分布在 0 的两侧。指标实际值比平均值大的,其评价值大于 0;实际值比平均值小的,其评价值小于 0[13]。

对于服务列表中不同服务器节点 $S_1, S_2, \cdots, S_i$ 中相同的因素种类 R_j,其原始数值指标可表示为 $S_1R_j, S_2R_j, \cdots, S_iR_j$,则 Z 分数的计算结果可用 $ZS_1R_j, ZS_2R_j, \cdots, ZS_iR_j$ 来表示。

令 $S_1R_j=x_{1j}, S_2R_j=x_{2j}, \cdots, S_iR_j=x_{ij}$,

$ZS_1R_j=y_{1j}, ZS_2R_j=y_{2j}, \cdots, ZS_iR_j=y_{ij}$,

则有:

$$y_{ij}=\frac{x_{ij}-\bar{x}}{s}$$

其中:

$$\bar{x}=\frac{1}{n}\sum_{i=1}^{n}x_{ij}$$

$$s=\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_{ij}-\bar{x})^2}$$

可以用向量 $Z_j=(y_{1j}, y_{2j}, \cdots, y_{ij})$ 表示对评价因素 j 的数值在所有节点上的 Z-score 规范化结果,其中 j 最大取值为评价因素集合 X 的长度,i 最大取值为集群服务列表中节点数量。

将 $Z_1, Z_2, \cdots, Z_j$ 在同一时刻的计算结果映射出多因素的评判矩阵:

$$R=\begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1j} \\ y_{21} & y_{22} & \cdots & y_{2j} \\ \vdots & \vdots & \vdots & \vdots \\ y_{i1} & y_{i2} & \cdots & y_{ij} \end{bmatrix}$$

对服务器 i 来说,可以根据矩阵 R 按行截取得到向量 $Res_i=(y_{i1}, y_{i2}, \cdots, y_{ij})$,$Res_i$ 为服务器 i 的实时性能向量。

2.3 Delphi 法资源量的确定

Delphi 法(德尔菲法)由 Dalkey 和 Helmer 于 1964 年发明并首先将其应用于军事预测分析[14]。通过此方法可以对智慧农业平台内的服务器集群和

各个微服务(如数据上传、图像采集和流媒体传送等业务)的逻辑性能需求进行有效的判断。此方法采用多轮征询背对背的通信方式征询专家组的意见,最后构造出可信度较强的预测意见供 DRF 权重算法使用。

服务器资源量(TR)、各个微服务所需资源量(N)均可采用 Delphi 法获得可观的评议结果。以各个服务器资源量(TR)的求解为例,具体步骤如下:

(1)确定各个影响因素的相对资源系数a_i,$i=1,2,\cdots,n$,其中 n 为影响因素的总个数。

(2)邀请各行业的一线开发人员、平台管理人员和第三方农业软件或硬件设备开发人员对所提供服务器集群的详细性能指标进行评估审阅,预先设定 1 个序列值$a_i \in \{1,2,\cdots,i\}$,评估人员对每台主机的评价只可非重复地使用 1 个序列值。征询问卷结构包括专家编号、服务器编号、因素编号、资源量最低值、资源量最高值、最可能资源量。

(3)统计分析,进行一致性检验。

$$\text{首先计算本次测试的应答率}(Call)=\frac{Answer}{Attend}$$

其中 $Answer$ 是完成全部上报的人数,$Attend$ 是调查对象的总人数。如果 $Call<90\%$ 则认为本次统计无效,返回步骤(2)。

若应答率符合要求,则继续计算变异系数(CV)。

变异系数又称离散系数、标准离差率或单位风险,其优势在于作为一个无量纲量,统计时不需要考虑数据的平均值[15]。对于每台服务器每种资源因素的最低、最高、最可能资源量,按照专家的打分分别统计其变异系数(CV):

$$CV=\frac{\sigma}{\mu}$$

其中 σ 为数据的标准差,μ 为数据的平均值。如果超过 40% 的数据变异系数大于 0.5,则将本轮全体评价结果反馈给专家,要求专家再次进行评价,直到满足要求为止。

(4)求出每台服务器的资源量。

假设 m 个完成上报的人员进行了 k 轮试验,对第 i 台服务器的第 j 种资源量(R_{ij}),可由以下公式求得:

$$R_{ij}=\frac{1}{k}\sum_{p=1}^{p=k}\frac{1}{m}\sum_{n=1}^{n=m}a_{npij}$$

$$a_{npij}=0.25\times\max(a_{npij})+0.25\times\min(a_{npij})+0.5\times prob(a_{npij})$$

其中 n 为上报人员的编号,p 为试验的轮数,$\max(a_{npij})$ 为资源量最高值,$\min(a_{npij})$ 为资源量最低值,$prob(a_{npij})$ 为最可能资源量。

2.4 DRF 权重算法的实现

DRF 算法是一种通用的多资源分配策略,其根据主导资源所占的最优份额来实现[16]。Ghodsi 等[17]提出在分布式集群中使用 DRF 算法实现集群调度具有更好的吞吐量和公平性,Fan 等[18]在 Hadoop YARN 中通过 DRF 算法增强了对网络带宽资源的分配管理,Meskar 等[19]将 DRF 算法引入移动边缘计算(MEC)领域,实现了公平分配多种资源。在用户有异质需求的场景环境中,可以采用 DRF 算法实现多种分配因素下的资源公平分配。对于智慧农业物联网以及数据终端来说,不同业务(数值分析、数据上传、图像采集、流媒体传送等)具有不同的性能需求偏好,因此需要实现对有不同需求的用户在选择节点时根据其性能特性有针对性的侧重。

设各个评价因素种类为$R_1,R_2,\cdots,R_n$,系统内资源总量用向量 $TR=(TR_1,TR_2,\cdots,TR_n)$ 来表示。对于客户端任务C_i来说,其所需要的因素资源量由向量$N_i=(C_iR_1,C_iR_2,\cdots,C_iR_n)$构成。

已知客户端任务C_i,其资源需求向量为N_i,资源总量为 TR,则客户端任务C_i的主导份额($MainResource$)计算公式为:

$$MainResource=\max(\frac{N_i}{TR})$$

对于上文 2.2 中的评判矩阵 R 和 2.3 中 Delphi 法确定的服务器资源量、各个微服务所需资源量(N),虽然已经求得对每个服务器节点不同性能评价指标的归一化数据,但在一线农业应用客户端选择节点的过程中,由于对于不同微服务在多资源环境中的各资源因素重要程度不一样,故影响力也不同,所以在综合评判时,必须给出各个因素在总评价中的重要程度,用权重集(A)来表示。

根据已求得的主导份额($MainResource$),则权重集(A)可以表示为:

$$A=(1,1,\cdots,\frac{1}{2})$$

这里的$\frac{1}{2}$为出现的 $MainResource$ 所在的权重位置。在每台设备的实时性能向量中,选中主导份额将其乘以$\frac{1}{2}$,即

$DRFRes_i = Res_i \times A$

式中，$DRFRes_i$ 为加权后的实时性能向量，Res_i 为各个服务器的实时性能向量。

通过以上处理可以达到放大该性能指标对最后评判标准影响的作用。

2.5 负载均衡算法整体流程

步骤 1：每当微服务服务器集群节点发生变化或平台上新增微服务时，数据终端客户端向 Eureka 更新服务列表，通过 Delphi 法确定服务器资源量（TR）、各个微服务所需资源量（N）。

步骤 2：从 Eureka 注册中心拉取服务节点列表，同时新建线程，获取客户端到服务端的响应延迟时间。

步骤 3：通过 Z-score 算法计算多因素评判矩阵（R）与各个服务器的实时性能向量（Res_i）。

步骤 4：通过 DRF 算法得到客户端业务的主导份额和权重集，从而得到加权后的实时性能向量（$DRFRes_i$）。

步骤 5：基于加权后的实时性能向量对每个服务器的性能作出综合评判，得到评判分数，服务器的编号用 j 表示，则综合评判分数记作：

$$Score_j = \sum_{k=1}^{n} (10 - r_k)^2$$

其中 r_k 为 $DRFRes_i$ 中不同种类的各个资源因素，n 为原因因素的种类数目。

步骤 6：将服务器的综合评判分数按照从小到大的顺序排序，选取前 50% 服务器，分别按照等差递减的出现比例随机选取。

步骤 7：每隔 5 min 返回步骤 2 重新执行。

3 性能测试与分析

3.1 试验环境

本试验构建了 1 个智慧农业物联网服务集群来模拟农田物联网设备和数据终端向服务器上传传感器数据（JSON 格式）的业务，其中包括 1 个 Eureka 服务注册中心和 5 个微服务集群。服务器的配置具体包括：

（1）Eureka 注册中心服务器：

处理器：Intel Xeon E5-2682 v4@ 2.5 GHz；

内存：32 GB DDR4；

操作系统：Windows server 2008；

地理位置：中国上海阿里云。

（2）微服务集群服务器（共 5 台）：

处理器：1 核；

内存：百度云，2 台为 1 GiB，其余 4 台为 2 GiB；

操作系统：Windows server 2008、Ubuntu 18.04；

地理位置：中国上海阿里云、中国上海腾讯云、中国广州百度云、中国保定百度云、中国彰化谷歌云。

（3）负载发送压力主机：

处理器：Intel Core i7-7700HQ@ 2.8 GHz；

内存：32 GB DDR4；

网卡带宽：1 000 Mbps；

操作系统：Windows 10 Pro 1909。

（4）客户端负载均衡服务主机：

处理器：Intel Pentium G4560@ 3.5 GHz；

内存：4 GB DDR4；

网卡带宽：1 000 Mbps；

操作系统：Windows 10 Pro 1909。

模拟数据上传传感器的业务采用 GET 方式通过本地主机的 URL（统一资源定位器）上传包含多种类型数据的 JSON 文本，数据报文的结构如下：

```
{"DataHead":{
    "Date":"2020-01-01",
    "Time":"20:00:00",
    "FarmId":"001",
    "DataType":"Rain"
},
"DataBody":{
"Temperature":"25",
"Humidity":"50",
"WindSpeed":"15",
"Light":"10",
"isRain":"False",
"Rainfall":"0"
}}
```

上述负载均衡逻辑使用 Java 语言通过继承 Ribbon 中的 ClientConfigEnabledRoundRobinRule 类重写了 public Server choose（Object key）方法逻辑，通过一系列试验验证本算法在客户端负载均衡中的性能表现。

3.2 试验内容

响应延迟体现了用户对服务性能的最直观感受，本试验设计基于 HTTP 访问微服务模拟农业传感器数据终端上传数据业务，从而测试试验过程中

的平均响应时间和吞吐量。通过将上述算法所实现的负载均衡逻辑与 Ribbon 自带的默认轮询算法、高可用算法和时间权重算法进行对比,从而比较本算法给用户体验带来的影响。

3.2.1 *平均响应时间测试* 对于不同并发量,采用 LOCUST 这一开源负载均衡测试工具[20]来测试平均响应时间,其完全基于事件驱动,使用 Gevent 提供的非阻塞 IO 和 Coroutine 来实现网络层的并发请求。基于不同的并发用户量进行单轮周期为 15 min 的测试,测试结果见图 3。

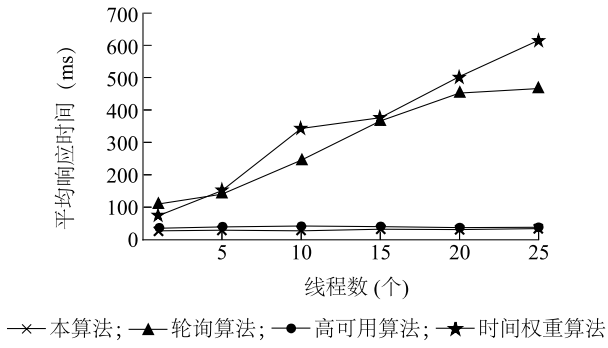

图 3 平均响应时间

Fig.3 Average response time

3.2.2 *TPS(每秒事务数)测试* *TPS* 是衡量系统性能的一个非常重要的指标[21],事务是指客户机向服务器发送请求直到服务器返回请求的过程。对于分布式系统来说,*TPS* 反映了系统的整体处理能力,采用 JMeter 测试工具,对 2 种不同的负载均衡方案的 *TPS* 各进行为期 15 min 的模拟农业物联网信息上传压力测试,结果见图 4。

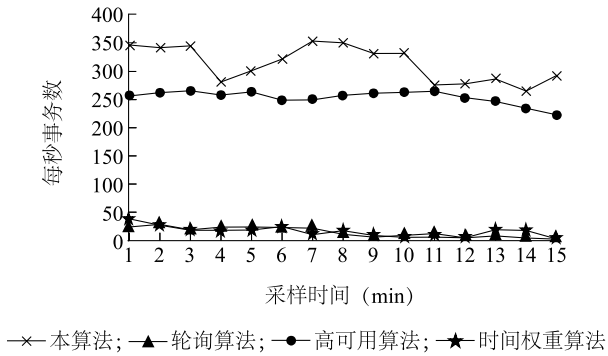

图 4 15 min 周期下的 TPS 测试

Fig.4 TPS test under the period of 15 min

3.3 试验结果分析

以上试验结果显示,基于改进 DRF 算法的微服务动态负载均衡算法不论是在不同并发线程数的响应时间上还是在压力测试下的平均响应时间方面都体现出了明显的优势。在平均响应时间的测试中,可以看到随着测试时间的推移,本算法的并发连接数始终能够保持在较高的水平,相比 Ribbon 自动的 2 种常用算法(轮询算法和时间权重算法),由于前 2 种算法在内存泄露和流回收方面的弱势,线程数增大时平均响应时间出现了较大幅度的提升,而本算法在使用 JAVA 语言实现时通过共用静态对象、优化代码、阻塞任务使用多线程并发执行、规避线程不安全代码等设计,减少了内存开销,能够在并发线程数不断增多时体现出稳定抗压的特点。

4 结 语

本研究通过提出一种基于改进 DRF 算法的客户端负载均衡策略,引入多角度的性能测评因素,通过 *Z*-Score 算法将测评因素标准化,进而结合 Delphi 算法计算出性能权重,从而在智慧农业平台的微服务化下实现服务高效负载均衡。通过本算法,实现不同地域、网络环境下的各个农业生产单位都可以以高性能、低耦合度和简易的方式接入系统平台,有效地提升客户端使用微服务的并发性能、吞吐性能,从而形成更加科学的负载均衡逻辑,在未来该技术可以与 5G 技术相结合,从而构建出高速农业微服务平台系统。本算法对后期研究中更多评价因素的引入和基于时间序列的流量峰值预测具有进一步的研究空间和价值,从而使每个用户获得高可用性的农业大数据和物联网微服务体验。

参考文献:

[1] 李道亮,杨 昊. 农业物联网技术研究进展与发展趋势分析[J].农业机械学报,2018,49(1):1-20.

[2] 李 菁. 改进快速稀疏算法的云计算资源负载均衡[J].微型电脑应用,2019,35(10):36-38.

[3] FOWLER M, LEWIS J. Microservices[EB/OL]. (2014-03-25)[2020-03-04]. https://martinfowler.com/articles/microservices.html.

[4] 辛园园,钮 俊,谢志军,等. 微服务体系结构实现框架综述[J].计算机工程与应用,2018,54(19):10-17.

[5] 龙新征,彭一明,李若淼. 基于微服务框架的信息服务平台[J].东南大学学报(自然科学版),2017,47(增刊 1):48-52.

[6] 李 瑾,郭美荣,高亮亮. 农业物联网技术应用及创新发展策略[J].农业工程学报,2015,31(增刊 2):200-209.

[7] 王 冉,徐本崇,魏瑞成,等. 基于无线传感网络的畜禽舍环境

监控系统的设计与实现[J].江苏农业学报,2010,26(3):562-566.

[8] 陈 栋,吴保国,陈天恩,等. 分布式多源农林物联网感知数据共享平台研发[J].农业工程学报,2017,33(增刊1): 300-307.

[9] 徐琛杰,周 翔,彭 鑫,等. 面向微服务系统的运行时部署优化[J].计算机应用与软件,2018,35(10):85-93.

[10] 辛园园,钮 俊,谢志军,等. 微服务体系结构实现框架综述[J].计算机工程与应用,2018,54(19): 10-17.

[11] 徐 浪,王青华. 描述统计学[J]. 成都:西南财经大学出版社,2001.

[12] 叶小榕,邵 晴. 基于Spark的大规模社交网络社区发现原型系统[J].科技导报,2018,36(23):93-101.

[13] 叶 青. 国家工业现代化水平的实证研究——基于二次现代化理论的视角[J].理论与现代化,2016(5): 15-23.

[14] SHELTON K, HAYNES C A, CREGHAN K A. Fundamentals of Delphi research methodology[M]. IGI Global: Hershey, 2018: 233-257.

[15] 程 果. 基于HTTP的动态自适应流媒体技术的研究[D].广州:广东工业大学,2017.

[16] 柯尊旺,于 炯,廖 彬. 适应异构集群的Mesos多资源调度DRF增强算法[J].计算机应用,2016,36(5):1216-1221.

[17] GHODSI A, ZAHARIA M, HINDMAN B, et al. Dominant resource fairness: fair allocation of multiple resource types[C]// USENIX. 8th USENIX Symposium on Networked Systems Design and Implemention. Boston, USA: USENIX, 2011:24.

[18] FAN X, LANG B, ZHOU Y, et al. Adding network bandwidth resource management to Hadoop YARN[C]// IEEE. IEEE 2017 Seventh International Conference on Information Science and Technology. Da Nang, Vietnam: IEEE, 2017: 444-449.

[19] MESKAR E, LIANG B. Fair multi-resource allocation with external resource for mobile edge computing[C]// IEEE. IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). Honolulu, USA: IEEE, 2018: 184-189.

[20] COULSON N C, SOTIRIADIS S, BESSIS N. Adaptive microservice scaling for elastic applications[J]. IEEE Internet of Things Journal, 2020, 7(5): 4195-4202.

[21] VARGAS-SANTIAGO M, MORALES-ROSALES L, POMARES-HERNANDEZ S, et al. Autonomic web services enhanced by asynchronous checkpointing[J]. IEEE Access, 2018(6): 5538-5547.

(责任编辑:陈海霞)