基于遗传算法的改进时序预测模型研究

李思莉,杨井荣
(成都理工大学 工程技术学院 电子信息与计算机工程系,四川 乐山 614000)

摘 要:云计算系统通过对存储、软件、服务等资源进行统一调度来为用户提供所需的服务。用户的需求具有多样性、多变性,使用弹性伸缩技术可以提高用户满意度,很好地解决资源利用率和应用系统之间的矛盾,是云计算的关键技术之一。然而,网络应用程序的工作负载通常是动态的,并且很难预测。因此,云计算中 Web 应用的关键技术是根据负载进行资源的动态分配,这是研究的热点,也是难点。目前,针对动态伸缩算法,提出的解决方案多是独立的、单一的或基于过去资源使用率进行提前预测。但这些方法容易导致资源利用不足。该文提出利用遗传算法改进时序预测模型 ARIMA 计算所需的虚拟主机数,以实现提高资源利用率,达到资源快速伸缩的目的。所提出的模型已经用几个基准工作负载进行了验证,在虚拟主机数和响应时间方面有一定的改善。

关键词:云计算;弹性伸缩;动态分配;遗传算法;ARIMA
中图分类号:TP301.6 **文献标识码**:A **文章编号**:1673-629X(2020)11-0084-05
doi:10.3969/j.issn.1673-629X.2020.11.016

Research on Improved Time Series Prediction Model Based on Genetic Algorithm

LI Si-li, YANG Jing-rong
(Department of Electronic Information and Computer Engineering, The Engineering & Technical College of Chengdu University of Technology, Leshan 614000, China)

Abstract: Cloud computing system provides users with the required services through the unified scheduling of storage, software, services and other resources. Users' needs are diverse and changeable. The use of elastic and scalable technology can improve user satisfaction and solve the contradiction between resource utilization rate and application system, which is one of the key technologies of cloud computing. However, the workload of network applications is usually dynamic and difficult to predict. Therefore, the key technology of Web application in cloud computing is the dynamic allocation of resources according to the load, which is the research hotspot and also the difficulty. At present, for dynamic scaling algorithms, most of the proposed solutions are independent, single or based on the past resource utilization to predict in advance. But these methods are easy to lead to insufficient utilization of resources. We adopt genetic algorithm to improve the time series prediction model to calculate the number of virtual hosts needed, so as to improve the utilization rate of resources and achieve the rapid resource expansion. The proposed model has been validated by several benchmark workloads, and has some improvement in the number of virtual hosts and response time.

Key words: cloud computing; elastic expansion; dynamic allocation; genetic algorithm; ARIMA

0 引 言

互联网技术的发展带来了电子商务等业务的迅猛发展,用户请求和互联网环境越来越复杂,服务器负载急剧增加,为互联网应用的高可用性、高性能和服务质量带来了严峻的挑战。传统的 Web 集群模式由于集群规模固定的问题,无法应对突发性访问激增带来的负载压力。

云计算技术的出现使得集群规模可以随工作压力变化而动态调整,资源按需获取成为可能。云计算中的弹性伸缩是一种云计算系统根据系统需求变化而自动调整的技术。它能有效提高用户满意度,特别是当系统需求不稳定时,能很好地调度资源,避免资源分配不足或资源浪费。其组织分类如图 1 所示。

对于资源的伸缩,一般分为水平伸缩和垂直伸缩。在水平伸缩(伸/缩)过程中,以虚拟机为最小资源单位按需要添加或释放虚拟机;在垂直缩放(上/下)过

收稿日期:2019-12-15 **修回日期**:2020-04-15
基金项目:乐山市科技计划项目(19JRK229);四川省教育科研项目(18ZA0071)
作者简介:李思莉(1974-),女,硕士,副教授,研究方向为云计算、软件系统架构。

程中,通过改变分配给已经运行的虚拟机的资源来实现,例如增加(或减少)分配的 CPU 或内存。对于弹性伸缩的实现机制,目前比较流行的是反馈触发策略[1]和提前预测。反馈触发策略是根据一些性能指标和预定义的阈值来触发伸缩,也就是说,系统将根据用户负载是否达到某个确定的阈值来决定增加或减少资源。但是,需要精确的定量值,通常容易出现不确定性[2]。同时,如果用户负载非常不稳定,就会造成系统资源的频繁伸缩,这本身也会给系统带来很大的负担。加之,这种反馈触发策略在资源分配部署过程中存在较大的延时问题,在用户负载大规模增加的情况下,处理能力有限。

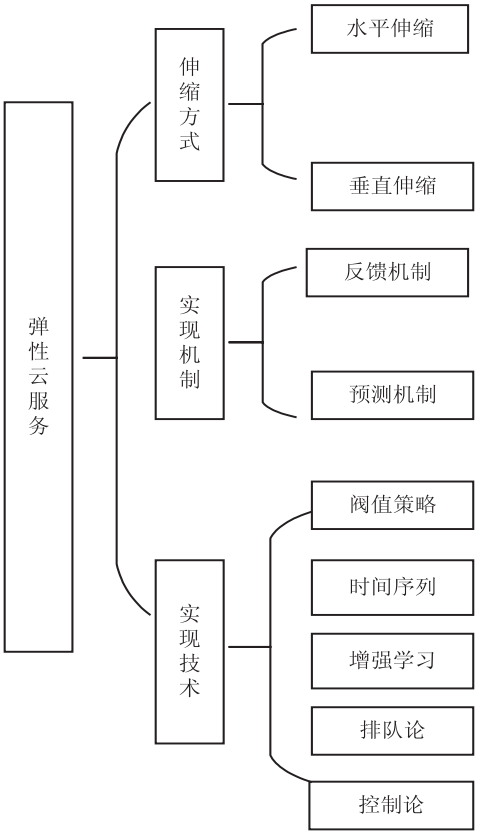

图 1　弹性云服务组织分类

预测模式以时间序列方法[3]、控制理论[4]、增强学习(RL)[5]和队列模型[6]为主,但单纯的时间序列方法难以预测峰值和工作负载真实变化。队列模型只适用于静态环境。此外,队列模型的现有方法(QM)对弹性系统有不切实际的假设,并且这些假设在云计算环境中并不适用,因为在云环境中,动态工作负载是一种常态。

该文利用遗传算法改进时序预测模型,结合当前资源使用率,对未来趋势进行预测,从而实现了资源动态快速伸缩。通过实验表明,该预测模型在同等虚拟用户请求的时序下,虚拟机数量在满足 SLA 的情况下较少,并且响应速度下降。

1　相关工作

1.1　排队论

排队论是排队等待中的概率处理最优化设计问题。排队论(QT)被广泛用于表示基于 internet 的应用程序和传统服务器性能指标,如队列长度和平均等待时间请求的时间。云应用程序场景使用一个简单的 QM 对于在“n”个虚拟机之间分配请求的负载平衡器。现实云环境中的自动缩放问题是通过周期性地改变传入的到达和无限服务器容量的假设。排队理论包括到达过程、排队规则、服务机制,如队列长度和平均等待时间请求的时间。文献[7]中使用排队论对工作负载和服务平均响应时间进行预测,从而实现资源的动态分配。目前,被广泛使用的排队模型是 M/M/1 和 G/G/1[8],M/M/1 将用户服务请求过程看作为是一个参数为 λ 的泊松流,其中到达时间和服务时间是基于指数分布的,而 G/G/1 到达时间和服务时间是基于一般分布的,这种模型最大的问题是当面对峰值负载时,可能造成大量的资源浪费。排队论适用于结构相对固定的系统,如果系统结构发生变化,需要重新建模。因此,排队论在云计算资源动态伸缩的管理中代价很高,通常不是一个太好的选择。

1.2　时间序列分析预测模型

基于时间序列分析的弹性策略是实现资源弹性管理的常用方法。它能够预测应用程序的未来需求,所以通常用来提前申请资源,从而有效避免了在虚拟机启动和应用程序部署上花费时间。这一策略具有巨大的潜力,在金融、工程、经济和生物信息学等领域有着广泛的应用。其独特的特性也决定了该方法同样适用于云计算环境下的资源预测。

目前,时间序列预测分析方法的主要模型有朴素模型(Native)、自回归预测模型(AR)、自回归滑动平均模型(ARMA)、差分自回归滑动平均模型(ARIMA)以及扩展指数平滑模型(ETS)和神经网络预测模型等[7]。朴素模型非常简单,它假设最后一次的观测值将出现在下一个时间间隔内。这个模型只需要单个时间点序列。

自回归预测模型可以根据最近几个时间段的负载值预测下一个时刻的负载。其线性表达式为:

$$y_t = c + \varphi_1 y_{t-1} + \varphi_2 y_{t-2} + \cdots + \varphi_p y_{t-p} + \varepsilon_t \quad (1)$$

其中,c 是常量,ε_t 是独立的误差项,φ_1 是自相关系数,该系数如果小于 0.5,其预测结果将极不准确。

自回归滑动平均模型 ARMA(p,q)包含了两个多项式的平稳随机过程,一个用于自回归,另一个用于移动平均线。其模型可以表示为:

$$X_t = c + \varepsilon_t + \sum_{i=1}^{p} \varphi_j x_{t-i} + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} \quad (2)$$

其中，p 和 q 是模型的自回归阶数和移动平均阶数；φ 和 θ 是不为零的待定系数；ε_t 是独立的误差项；X_t 代表平稳、正态、零均值的时间序列。ARMA 模型适用于预测在一定趋势内变化的时间序列。

差分自回归滑动平均模型 ARIMA（p,d,q）是 ARMA 模型的扩展，它不直接考虑其他相关随机变量的变化，其模型可以表示为：

$$\left|1 - \sum_{i=1}^{p} \varphi_i L^i\right| (1 - L)^d X_t = \left|1 + \sum_{i=1}^{q} \theta_i L^i\right| \varepsilon_t \quad (3)$$

其中，p 是自回归项数，q 是滑动平均项数，d 是使之成为平稳序列所做的差分次数（阶数）。该模型的缺点是参数选取困难。

神经网络预测模型适用于非线性时间序列的预测，它采用梯度下降法来计算阈值和权值，并通过不断修正网络权值和阈值使误差函数沿负梯度方向下降，逼近期望输出。误差函数表达式为：

$$E = \frac{\sum_{i}^{n} (t_i - o_i)^2}{2} \quad (4)$$

其中，t_i 为期望的输出，o_i 为网络实际输出。但是，神经网络预测模型容易导致神经网络陷入局部最优解，降低预测精度。

1.3 时间序列预测算法

目前，时间序列预测算法主要以差分自回归滑动模型（ARIMA）[9] 为基础，这些算法对于时间有较强依赖性的线性数据有较好的预测效果，但对于非线性数据，如果数据波动过大，将对非平稳序列差分进行平稳化处理，变成平稳序列之后再进行移动回归，这将会导致预测误差增大。在实际云环境中，单一的 ARIMA 算法很难准确预测工作负载。因此，目前有些研究人员已经对 ARIMA 算法进行了改进，如文献[10]用傅里叶变换提高预测准确度；文献[11]中引入平均误差提高预测精确度；文献[12]对下一个预测周期进行预测误差补偿，防止误差进一步扩大；文献[13]对历史窗口大小进行动态调整。这些算法是以 ARIMA 算法为基础进行的改进，对提高预测准确度有一定的帮助。

2 改进的时序预测算法

2.1 资源分配弹性伸缩框架

由于并发或重复的请求是从多个客户机发出的，因此到达间隔时间成为一个随机变量，又由于到达的请求遵循具有无限计算能力的泊松过程，所以系统使用 M/M/ m 队列模型来处理到达的服务请求，m 是提供的服务器数量。仿真结果表明，稳态下队列长度等待的平均时间会随平均间隔时间的增加而降低，如果增加服务器 m（虚拟机）的数量，等待时间会降低。但是，在云计算环境下，需要通过负载平衡策略和时间预测算法计算出在不违反 SLA 原则的基础上，最小化资源的占用率，即 m 的值。其框架示意图如图 2 所示。

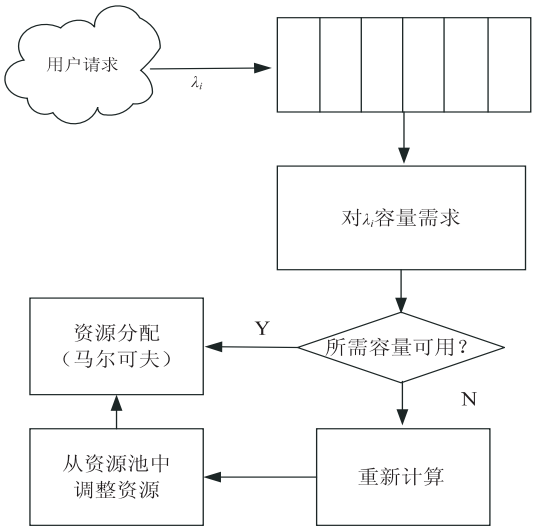

图 2 资源分配弹性伸缩框架

从图 2 可以看出，弹性分配策略主要分为两个阶段，第一阶段负责预测下一个时间间隔的需求，第二阶段根据改进的基于时间序列的算法求取最小资源数，动态调整调度资源。

2.2 改进的基于遗传算法时序的预测算法 GA-ARIMA

本算法是对 ARIMA 基本算法的改进，因此在设计时首先使用了 arima( ) 函数创建自回归模型，接着使用 auto. arima( ) 函数根据确定的时间间隔调整参数，最后使用 ets( ) 函数调整平滑指数，使之与数据的拟合度最高。其具体算法伪代码如下：

```
input:client request      //用户请求
output:next point    //下一个时间间隔需求预测
1:init( );
2:foreach timeInterval do
3:TS<-read(information);   //将用户请求转换为时序
4:   foreach m do
5:value[m]<-predictAhead(m);
6: end
7:nextValue<-UsingPerviousValue(value);
8:end
```

该算法首先完成初始化的工作，然后对于每个时间间隔（这个值可以设定），将用户请求（可以多个，高并发）转换为时序，根据时序选择适用的参数及模型，得到需求值。最后根据前一个需求值等到下一个预测值。

UsingPerviousValue 函数需要用到遗传算法[14]，该算法模拟达尔文的进化论，其具体描述如下所示：

```
input:perviousValue
```

output:nextValue
1:Initializate ();
2:While(not Terminate-Condition) do
3:evalPopulation();
4:select();
5:crossSelected();
6:mutateResultValue();
7:evaluateResultValue();
8:updatePopulation();
9:end
10:outputNextValue;
11:end

2.3　资源分配

根据 2.2 描述的预测算法得到的结果,使用 M/M/ m 队列模型[15]来分配资源,其系统资源利用率根据等式(5)来计算。

$$\rho = \frac{\lambda}{m\mu} \tag{5}$$

其中,ρ 是系统利用率,λ 是到达率,μ 为处理率,m 为服务器的数量。文中的目标是找到满足用户需求的 m 的最小值。因此变换式(5)得:

$$m = \lceil \frac{\lambda}{\rho\mu} \rceil \tag{6}$$

M/M/ m 模型中,系统利用率 ρ 和响应时间 t 密切相关,响应时间 t 计算如下:

$$t = \frac{\mu^{-1}}{1-\rho} \tag{7}$$

将式(5)和式(7)带入式(6)得:

$$m = \frac{t\lambda}{\lambda\mu - 1} \tag{8}$$

根据式(8)就可以计算出所需资源,要使 m 值最优,采用如下算法:

输入:预测值(算法 2 的结果)
输出:m
初始化;
1:foreach(TS) do
2:λ <-getForecastPrevious(model);
3:μ <- getProcessingRate();
4:sla<-getMaxRT();
5:m <-calculateResources(λ , μ ,sla);
6:end

通过预测获得了下一个时序的到达率后,每个虚拟机和最大响应时间的 μ 值可以计算出来,最后带入式(8)得 m 的值。

3　仿真实验分析

为了验证提出的基于遗传算法改进时序预测的云计算弹性伸缩策略的有效性,利用基于 OpenStack[16]开源项目搭建云平台,将 CPU 需求(虚拟机数量)作为实验数据,在 Matlab 上进行仿真,初始参数设置如表 1 所示。

表 1　参数设置

参数	初始值	变化的值
虚拟用户请求	随机	随机
处理率(μ)	每秒 10 个请求	每秒 20 个请求
SLA 值	0.4 秒	0.7 秒

图 3 模拟了 400 秒内虚拟用户的随机请求时序。

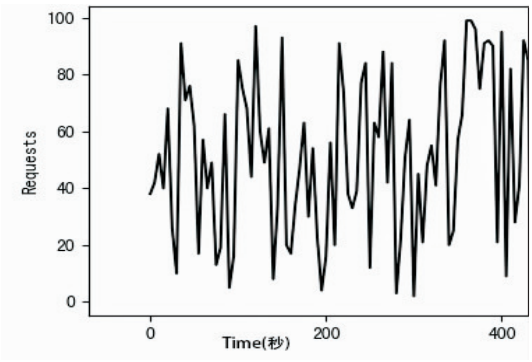

图 3　时序图

根据计算公式和实验数据得出的误差率如表 2 所示。

表 2　各模型误差值

预测模型	平均绝对误差(MAE)	均方根误差(RMSE)	平均绝对百分比误差(MAPE)
AR(1)	1.37	1.94	16.90
ARMA(1,1)	1.32	1.88	16.41
ARIMA(1,0,1)	1.28	1.84	16.07
GA	1.27	1.82	16.05

由实验结果可以看出,在同一时序下,利用遗传算法改进的预测模式平均绝对误差、均方根误差等都有一定程度的降低。

根据式(8)计算所需虚拟机数量,在一周的时间内,相同负载下的不同模型实验结果如图 4 所示。

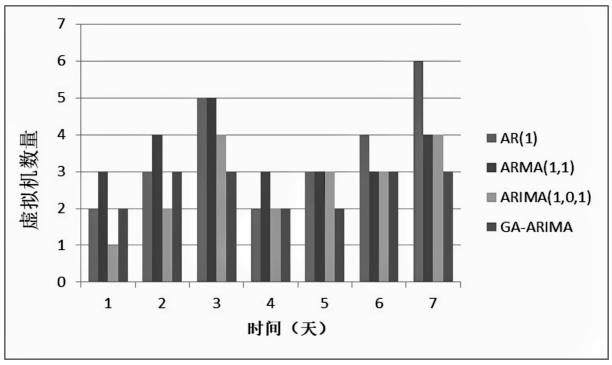

图 4　虚拟机数量对比

4　结束语

首先介绍了云计算弹性伸缩策略的必要性和现存

策略的弊端,对目前现有的预测模型进行了相关的介绍,指出它们的优劣。在此基础上利用遗传算法对 ARIMA 模型进行改进,并对未来一段时间的虚拟机需求做预测,从而决定是否需要伸缩。通过实验表明,提出的基于遗传算法的预测模型在一定程度上减少了 CPU 使用率,减少了预测误差,有一定的优越性。

参考文献:

[1] KOPEREK P,FUNIKA W. Dynamic business metrics-driven resource provisioning in cloud environments[C]//International conference on parallel processing and applied mathematics. Czestochowa,Poland:Springer-Verlag,2017:171-180.

[2] AL-HAIDARI F,SQALLI M,SALAH K. Impact of CPU utilization thresholds and scaling size on autoscaling cloud resources[C]//5th international conference on cloud computing technology and science (CloudCom). Bristol, United Kingdom:IEEE,2013:256-261.

[3] KIM K I,WANG W,QI Y,et al. Empirical evaluation of workload forecasting techniques for predictive cloud resource scaling[C]//2016 IEEE 9th international conference on cloud computing. San Francisco,CA:IEEE,2016:1-10.

[4] ELDIN A A,TORDSSON J,ELMROTH E. An adaptive hybrid elasticity controller for cloud infrastructures[C]//2012 IEEE network operations and management symposium. Maui:IEEE,2012:204-212.

[5] SALAH K. A queueing model to achieve proper elasticity for cloud cluster jobs[C]//IEEE Sixth international conference on cloud computing. Santa Clara:IEEE,2013:755-761.

[6] DUTREILH X,KIRGIZOV S,MELEKHOVA O,et al. Using reinforcement learning for autonomic resource allocation in clouds:towards a fully automated workow[C]//Seventh international conference on autonomic and autonomous systems. Venice:IEEE,2011:67-74.

[7] URGAONKAR B B,SHENOY P,CH A,et al. Agile dynamic provisioning of multi-tier internet applications[C]//International conference on autonomic computing. Washington DC:ACM,2010:217-228.

[8] TESAURO G,JONG N K,DAS R,et al. A hybrid reinforcementlearning approach to autonomic resource allocation[C]//IEEE international conference on autonomic computing. Dublin:IEEE,2006:65-73.

[9] FANOODI B,MALMIR B,JAHANTIGH F F. Reducing demand uncertainty in the platelet supply chain through artificial neural networks and ARIMA models[J]. Computers in Biology and Medicine,2019(2):33-45.

[10] VAZQUEZ C,KRISHNAN R,JOHN E. Time series forecasting of cloud data center workloads for dynamic resource provisioning[J]. Journal of Wireless Mobile Networks, Ubiquitous Computing,and Dependable Applications,2015,6(3):87-110.

[11] ZHANGG P. Time series forecasting using a hybrid ARIMA and neural network model[J]. Neurocomputing,2003,50:159-175.

[12] MESSIAS V R,ESTRELLA J C,EHLERS R,et al. Combining time series prediction models using genetic algorithm to autoscaling Web applications hosted in the cloud infrastructure[J]. Neural Computing & Applications,2016,27(8):2383-2406.

[13] EL DESOUKY A A,ELKATEB M M. Hybrid adaptive techniques for electric-load forecast using ANN and ARIMA[J]. IEE Proceedings:Generation,Transmission and Distribution,2015,147(4):213-217.

[14] HOLL J H. Genetical algorithms[J]. Journal of Mechanical Engineering Science,1992,267(1):66-73.

[15] 许浩然,刘广钟. 基于排队论的工业无线传感网超帧结构研究[J]. 计算机技术与发展,2019,29(1):6-10.

[16] 王 元,王志明. OpenStack 云平台的监控系统算法设计与实现[J]. 计算机技术与发展,2018,28(7):196-199.