基于ARIMA模型的虚拟资源动态调度方法

杨冬菊 邓崇彬
(大规模流数据集成与分析技术北京市重点实验室 北京 100144)
(北方工业大学云计算研究中心 北京 100144)

摘 要 将应用部署到云端已经成为业界越来越普遍的做法,高并发、大流量已经成为多数云应用的一大特征。如何应对不断增长的高并发和用户流量的激增、合理利用资源、保障应用的稳定运行是云资源管理需要解决的重要问题。针对基于监控数据进行资源调整的方式容易引发资源调整滞后的问题,提出了一种基于ARIMA预测模型进行资源调整的虚拟资源动态调度方法。该方法能够根据预测的请求量,结合当前资源的负载能力来计算所需的资源规模,从而进行虚拟机资源的配置或释放。实验结果表明,所采用的预测模型能够较好地拟合实验的场景,通过使用基于预测模型的资源调度算法能够及时、有效地保证云服务质量。

关键词 云应用,流量激增,服务质量,预测模型,资源动态调度
中图法分类号 TP301 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2017.10.003

Dynamic Scheduling Method of Virtual Resources Based on ARIMA Model

YANG Dong-ju DENG Chong-bin
(Beijing Key Laboratory on Integration and Analysis of Large-scale Stream Data, Beijing 100144, China)
(Research Center for Cloud Computing, North China University of Technology, Beijing 100144, China)

Abstract Deploying applications to the cloud has become an increasingly common practice in the industry, high concurrency and high traffic have become major features of most cloud applications. How to deal with the rising of the high concurrency and the surge of user traffic, to use resources reasonably, and to ensure the stable operation of the application, are important issues to solve for the cloud resource management. Considering the adjustment of resources based on monitoring data is easy to trigger the delay of resource scheduling, a dynamic scheduling method for resource adjustment based on ARIMA prediction model was proposed in this paper. The method can calculate the required resource size according to the demand of the forecast and the load capacity of the current resources scale, thus configurating or releasing the virtual machine resources. The experimental results show that the prediction model can fit the scene well. By using the predictive model, the resource scheduling algorithm can effectively guarantee the quality of cloud services in a timely and effective manner.

Keywords Cloud application, Surge in traffic, Quality of service, Prediction model, Dynamic resource scheduling

1 引言

云计算为用户提供了一种按需分配和可扩展交付的模型[1],它将资源作为一种服务提供给用户,越来越多地被政府、研究机构和行业的领跑者用于解决复杂的计算和存储问题[2]。有效的虚拟资源调度能够促进云资源的合理利用,保障云应用的稳定运行,从而提高资源利用率,满足用户需求。

近几年,越来越多的云应用呈现出高并发、大流量的特征[3],如在国家科技计划项目申报系统中,用户的访问时间和访问量呈现出明显的峰值变化,峰谷的用户访问量可能会是峰底用户访问量的几百甚至几千倍;学校的学生选课系统、社团活动系统等都具有类似特征。如何应对不断增长的高并发和用户流量的激增、合理利用资源、保障应用的稳定运行是云资源管理要解决的重要问题。云资源调度问题通常被认为是一个组合优化问题,也是NP完全问题[4]。随着应用的访问量和数据量等的不断变化,虚拟机的负载也在变化,为了应对不断增长的高并发和突发的用户流量,往往会过度分配虚拟资源[5],这必然导致资源的利用率非常低下。为了更好地操作和管理资源,常采取监控措施[6]。当流量激增时,大多数应用通常会通过监控数据判断并预警系统超负荷运行,从而进行资源调整。这种通过监控预警调度资源的方式往往在时间上具有一定的延迟性。而将预测与监控相结合来进行资源调

到稿日期:2016-08-05 返修日期:2016-12-02 本文受北京市自然科学基金重点项目:面向大规模流数据处理的数据空间理论和关键技术(4131001)资助。

杨冬菊(1975-),女,博士,副研究员,主要研究方向为服务计算、行业资源中心关键技术、创建模式及其在领域中的示范应用,E-mail:yangdongju@ncut.edu.cn;邓崇彬(1990-),男,硕士,主要研究方向为虚拟资源调度,E-mail:dengchongbin@foxmail.com。

度的方法通常可以有效地解决上述问题。

本文基于 ARIMA 模型来预测应用的负载，在预测数据的基础上结合监控到的节点负载情况进行综合预判，完成虚拟资源的提前申请和部署，从而有效避免资源调度的滞后及应用的性能抖动，保证云应用的服务质量。

本文第 2 节对目前国内外在虚拟资源调度方面的相关工作进行概述；第 3 节对实验的整体架构进行介绍；第 4 节对选用的几种预测负载的方法进行介绍；第 5 节描述调度算法；第 6 节通过具体的实验对方法进行验证；最后总结所用方案的优缺点。

2　相关工作

随着云应用和云服务的快速发展，云上的虚拟化资源调度和分配是研究的主题[7]，如何利用调度策略来有效地降低执行的成本并提高资源利用率是研究的热点。

目前的虚拟资源调度策略一方面集中于资源管理框架的研究[8]，另一方面集中于调度策略和算法的研究[9-15]。在调度方法上，比较常用的是基于监控的资源调度[9-13]，也有一些团队提出基于预测的资源调度[14-15]。

为应对流量突增对系统带来的巨大压力，文献[8]提出了一套动态资源管理的框架。

文献[9]讨论了增量式和综合布局两种虚拟机放置方法，为节约资源提出了一个新的虚拟机放置策略。

文献[10-11]重点关注仿生算法中的蚁群算法、遗传算法以及粒子群算法，从而优化了虚拟机资源的分配。

文献[12-13]重点关注通过提高服务质量来调度虚拟资源。文献[12]提出了服务质量敏感的资源分配机制——基于那什谈判解的虚拟资源动态调度方法。文献[13]提出了基于资源配置技术的服务质量指标。

文献[14-15]重点关注预测模型对资源调度的作用。文献[14]提出了基于马尔可夫链的分析模型，并用该模型来预估在一个特定 SLO(Service Level Object)指标下所需要分配的虚拟机实例最少的数量，但该文侧重强调负载均衡器对于实现适当弹性服务的作用，没有考虑服务所需资源减少时的情况；文献[15]利用贝叶斯模型来预测 CPU 和内存密集型应用短期和长期所需要的虚拟资源，但对资源的需求主要集中在对 CPU 和内存的探讨上，没有充分考虑虚拟机的整体情况。

文献[16-17]通过大数据分析做预测，也具有一定的借鉴意义。

综上所述，在虚拟资源调度策略中考虑以预测负载的方式来调度资源的相关工作还很少，尽管一些预测虚拟资源需求的模型已经被提出，但它们考虑的方面相对单一。因此，本文借鉴现有的资源调度成果提出一种预测与监控相结合的虚拟资源调度方法，目的是动态调整不同需求下的资源，在不同的负载情况都能保证系统的平稳运行，降低资源消耗，满足用户的需求，保证服务的质量。

3　系统架构

本文所使用的系统架构(见图 1)包含如下几个部分：负载均衡模块、Web 服务模块、数据收集模块、ARIMA 模型预测模块、动态调度模块、基础设施模块。

(1)负载均衡模块：所有用户提交的请求都将通过负载均衡器转发到应用的 Web server 服务器，实验系统选用 Nginx 作为系统架构中的负载均衡服务器。

(2)Web 服务模块：通过负载均衡模块之后，用户的请求都将到达 Web server，接收到请求的服务器将处理后的结果返回给用户。

(3)数据收集模块：该模块主要通过实时监控来获得数据，通过实时监控收集系统的 CPU、内存、请求量等数据，并将收集到的数据进行预处理，将每秒的请求量存入数据库以供将来分析历史数据时使用。同样，通过该模块能够简单计算每个请求从发出到处理完成的时间，这为评估系统响应时间做了准备。

(4)预测模块：该模块通过接收数据收集模块中预处理的数据来计算系统在下一个时间间隔将要收到的用户请求数量，这里采用的时间间隔是 5min。一方面本文主要研究短期内系统负载的预测；另一方面，过高的频率统计和计算数据会对服务器产生一定压力。一般每台虚拟机从创建到部署可以在 30～96s 完成[13]，因此选择这个时间间隔足以应对即将给系统带来的访问压力，也可以根据实际需求调整时间间隔。

(5)动态调度模块：通过预测模块得到系统请求量的预测值及当前系统的响应时间，并判断是否满足预定的最低响应时间，从而决定增加或者减少虚拟机资源。

(6)基础设置模块：该模块主要包括物理服务器和部署在物理服务器上的虚拟机资源，为应用提供底层资源。

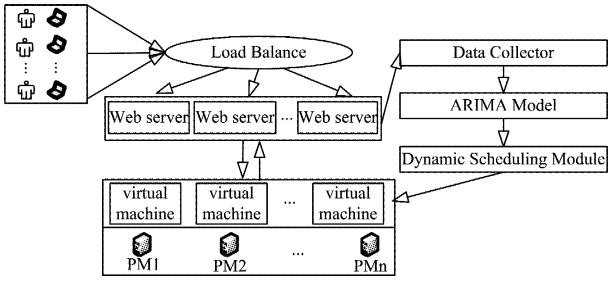

图 1　系统架构图

4　预测负载的方法

本节将讨论几种预测负载的方法，其中，应用负载主要指用户请求的数量，同时考虑到特定的 SLO，假定请求的响应时间在 2s 内。所要讨论的几种预测方法需要的输入数据均是每个时间间隔的请求量，通过方法得到的结果数据是下一个时间间隔的请求量，也即预测负载。此处主要讨论 3 种常用的预测方法：移动平均法、多项式拟合法、ARIMA 模型法。实验还引入了误差来分析 3 种方法中哪种更适合具有前述特点的应用。

表 1 列出文中所使用的符号及其定义。

表 1　符号定义

符号	描述
R	请求量，在本文中与负载通用
R_t	第 t 个时间间隔的负载
R_t^{pre}	第 t 个时间间隔的预测负载值
T/min	系统运行时间

(1)移动平均法:移动平均法是一种被广泛使用的技术指标,常用来以时间序列的方式预测未来数据[18]。在统计学中,移动平均法是通过创建一系列全数据集的不同子集的平均值来分析数据点的计算方法。它主要包括简单移动平均、累积移动平均及和加权移动平均[19],本文采用简单移动平均。

根据所定义的符号,请求所组成的序列可表示为 $R=\{R_1,R_2,\cdots,R_t\cdots|t\leqslant T/5\}$。根据数学中对移动平均的定义,可以预测 $t+1$ 时刻的请求量为:

$$R_{t+1}^{pre}=(R_1+R_2+\cdots+R_t)/t \tag{1}$$

其中,R^{pre} 为 $t+1$ 时刻用户请求量的预测值,t 表示第 t 个时间间隔,也是平均移动的周期,R_1-R_t 为前 t 个值。

(2)多项式拟合法:多项式回归分析是一种确定两个或两个以上变量间的相互依赖的定量关系的方法[20]。线性关系并不能很好地描述在每个时刻用户对应用的请求量,因此将多项式拟合的方法选作预测方法也是值得考虑的一个方向。

假定所预测的值与之前所统计的 t 个值符合某二次曲线关系,即:

$$R_t=at^2+bt+c \tag{2}$$

则只需要使用 3 组值就能够求出 a,b,c 的解集,然后再通过以下公式来预测第 $t+1$ 个时间间隔的请求量。

$$R_{t+1}^{pre}=a(t+1)^2+b(t+1)+c$$

(3)ARIMA 模型:自回归移动平均模型(Autoregressive Integrated Moving Average Model,ARIMA)基于马尔可夫的随机过程而建立,既吸收了回归分析的动态性优点,也保留了移动平均的优点[21]。非季节性的 ARIMA 模型使用 $ARIMA(p,d,q)$来表示,其中 p,d,q 均为非负整数,p 表示自回归模型的阶,d 表示差分的程度,q 表示移动平均模型的阶。季节性的 ARIMA 模型使用 $ARIMA(p,d,q)(P,D,Q)m$ 来表示,m 指每个季节的周期数,P,D,Q 分别指自回归、差分和移动平均[22]。

5 调度算法及其实现

本节是资源调度的核心部分,主要包括配置虚拟资源的虚拟资源动态调度算法(Virtual Resource Dynamic Scheduling,VRDS)和销毁虚拟资源的销毁算法。

5.1 虚拟机的资源配置算法及其实现

算法参数说明及虚拟资源配置流程如表 2、图 2 所示。

表 2 算法参数说明

符号	描述
$R_{current}^{max}$	当前资源规模下所允许承受的最大负载
$R_{current}^{min}$	当前资源规模下所允许承受的最小负载
R_i^{min}	Web 服务器 i 允许的最小负载值,低于该值则认为是空闲
R_i^{max}	Web 服务器 i 允许的最大负载,大于或等于该值则认为负载过重
Web	Web 服务器集合
Web_i	Web 服务器 i
N_{web}	Web 服务器数量
N_{need}	需要创建的虚拟机数量
N_{pm}	物理服务器数量
PM	物理服务器集合
$PMAscList$	满足创建虚拟机条件的按照负载升序排列的物理机集合
TH	阈值,即预测负载值达到最大/小负载值得次数
TH_{max}	设定的阈值的最大值

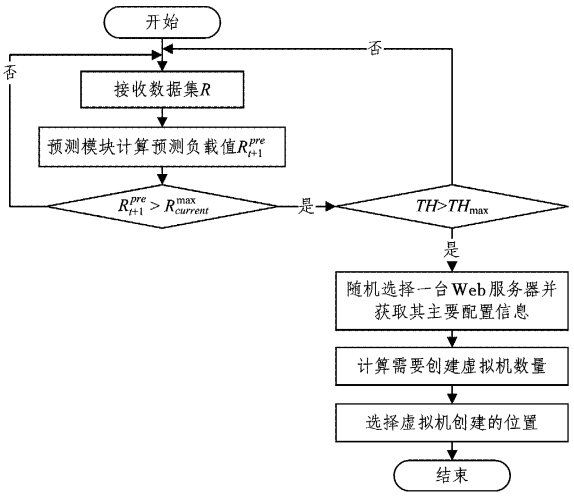

图 2 虚拟资源配置流程

数据收集模块首先收集原始数据,即用户每秒对应用产生的请求量,然后对其进行预处理。预处理主要包含两方面:1)剔除掉收集到的异常数据;2)将每秒的请求量聚集成每个时间间隔的请求量。实现的方案如下:

1)将经过预处理后的请求数量数据集 R 传递给预测模块。

2)预测模块利用 R 来预估下一个时间间隔的负载 R_{t+1}^{pre}。

3)判断 R_{t+1}^{pre} 与当前规模下的虚拟机所能承受的最大负载 $R_{current}^{max}$ 的关系。如果 R_{t+1}^{pre} 大于 $R_{current}^{max}$,则进行 TH 的判断,否则重复 1)和 2)。

4)如果 TH 小于 TH_{max},则重复过程 1)-3),否则开始启动资源的配置。

5)从正在运行的 Web 服务器中随机选择一台 Web_i,并获取其主要的配置信息(内存、CPU 核数、最大负载等)。

6)根据 R_{t+1}^{pre} 和 $R_{current}^{max}$ 来创建所需要的虚拟机数量,通过以下公式来计算所需要创建的虚拟机数量。

$$R_{current}^{max}=\sum_{i=1}^{N_{web}}R_i^{max} \tag{3}$$

$$N_{need}=(R_{t+1}^{pre}-R_{current}^{max})/R_i^{max} \tag{4}$$

7)根据 N_{need} 选择虚拟机创建的位置。首先从 $PM=\{PM_1,PM_2,\cdots,PM_i\cdots|i\leqslant N_{pm}\}$ 中选出满足创建虚拟机条件的物理机,然后通过对负载大小做升序排序形成 PMAscList。从 0 到 N_{need} 对 PMAscList 的数量进行取余运算,从而保证优先选择负载较小的物理机,进而确定需要创建的虚拟机所在的物理机的 IP。

虚拟资源配置的具体实现如算法 1 所示。

算法 1 虚拟资源的配置

输入:R_{t+1}^{pre},N_{web},$R_{current}^{max}$,PM,Web

输出:Position,即需要增加的虚拟机所在的物理机的位置

```
1. while true do
   //判断预测负载与当前最大负载的关系
2. if R^pre_t+1 >= R^max_current && TH > TH_max then
   //随机选择一台正在运行的 Web 服务器
3.     Randomly select Web_i from Web
       //获取该 Web 服务器的最大负载值
4.     R_i^max = Web_i.getMaxWorkLoad()
       //计算所需创建的虚拟机数量
```

```
5.      N_need=(R^pre_t+1 − R^max_current)/ R^max_i
        //返回创建虚拟机的位置
6.      return addVMPos(PM,N_need)
7. else
8.      break;
9. end while
10. function addVMPos(PM,N_need)
    //遍历物理机，寻找满足创建虚拟机条件的物理机
11. for pm in PM do
12.   if(pm. currentWorkload<pm. higestworkload()&&
      pm. hasResource()) then
13. list. add(pm)
14. end for
    //按照物理机的负载情况进行升序排序
15. PMAscList=sortByAsc(list)
16. size=PMAscList. size()
17. for i=0;i< N_need;i++ do
    //通过取余来选择虚拟机创建的位置
18. positions. add(PMAscList. get(i%size))
19. end for
20. return positions
```

5.2　虚拟机的销毁算法及其实现

当应用对资源的需求不再处于峰值时，如果依然保持峰值时需要的资源规模，显然是对资源的一种浪费。当一定时间内资源的负载已经小于或者等于虚拟机所设定的最小负载时，可以启用虚拟机（即 Web 服务器）的销毁算法。

虚拟机的销毁流程如图 3 所示。

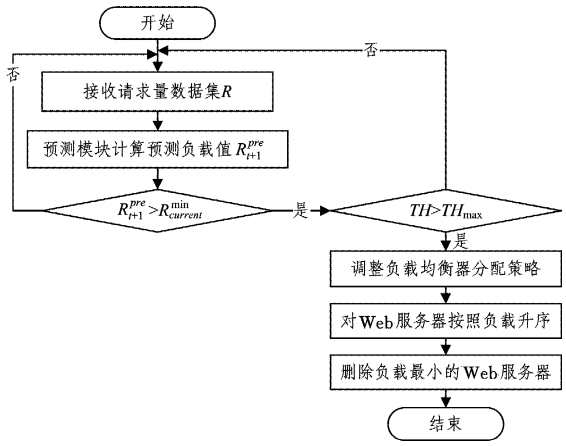

图 3　虚拟机销毁流程

实现方案如下：

1）将经过预处理后的请求量数据集 R 发送到预测模块。

2）预测模块计算预测负载值 R_{t+1}^{pre}。

3）判断 R_{t+1}^{pre} 与 $R_{current}^{\min}=\sum_{i=1}^{N_{web}}R_i^{\min}$ 的关系。如果 R_{t+1}^{pre} 小于或等于 $R_{current}^{\min}$，则继续重复过程 1）－3）。

4）如果 R_{t+1}^{pre} 大于 $R_{current}^{\min}$，则判断 TH 是否大于 $TH_{\max}$。

5）如果 TH 小于或等于 $TH_{\max}$，则重复过程 1）－4）。

6）如果 TH 大于 $TH_{\max}$，则调整负载均衡器的分配策略，将轮询分配的策略调整为权重分配策略，并将负载优先集中到一台服务器上。

7）对 Web 服务器进行负载值的升序排序，然后销毁掉最小负载值的 Web 服务器。

虚拟机销毁的具体实现如算法 2 所示。

算法 2　虚拟机销毁

```
输入：R^pre_t+1，R^min_current，Web
输出：销毁虚拟机成功(true)或者失败(false)
1. while true do
2. if(R^pre_t+1<=R^min_current)&&(TH>TH_max) then
3.   Adjust load allocation policy
   //获取删除虚拟机的位置
4.      rmPos=delVMPos(Web)
        //返回虚拟机是否销毁成功
5.      return destoryWeb(rmPos)
6. else
7.      break;
8. end while
9. function delVMPos(Web)
        //对虚拟机负载进行升序
10.     sortByAsc(Web)
        //获取负载最小的虚拟机位置
11.     return Web. get(0)
```

6　实验设计及结果

6.1　实验环境

为了验证预测模型与资源调度算法是否有效，搭建了一套实验环境来对本文提出的调度方法进行实验验证。所使用的实验环境为：3 台物理机（PM）（在其上搭建了 2 台 LoadRunner（LR）服务器），1 台负载均衡服务器（LB），4 台应用服务器，1 台 MySQL 服务器，具体的配置情况如表 3 所列。

表 3　实验资源的配置情况

名称	CPU	内存/GB	硬盘/GB
LR1	2 核 2.0GHz	4	130
LR2	2 核 2000MHz	4	100
LB	1 核 2000MHz	2	100
VM1	1 核 2000MHz	2	100
VM2	1 核 2000MHz	2	100
VM3	2 核 2600MHz	2	100
VM4	1 核 2000MHz	2	100
MySQL	2 核 2.67GHz	4	100
PM1	48 核 2.6GHz	64	830
PM2	48 核 2.6GHz	32	550
PM3	48 核 2.6GHz	32	550

6.2　预测方法的实验内容和结果分析

在上述实验环境中做了两组实验，一组采用默认的策略，另一组采用本文所提出的策略。使用 2 台 LoadRunner 服务器来模拟请求，然后通过前述预测模型进行预测。下文将通过实验来展示应用的请求数量，以及使用第 4 节中的预测方法得到的负载预测值与应用所记录的应用请求量的真实值形成的曲线图。

图 4 是实验所模拟的应用接受请求的总体趋势图，应用的请求数量从少到多逐渐增长，持续一定的时间之后，请求量会逐渐下降，这也是前述应用中需要解决的问题。图 5 给出了使用移动平均的方法来拟合应用的负载的情况，拟合的整

体趋势很好，但从图中可以发现，基于该方法的拟合曲线所得到的结果比较滞后，这并不能很好地帮助应用预测未来请求量的走势。图 6 中的方法在请求量增加和递减时都能够很好地拟合应用的负载，但在请求量处于峰值时段时预测结果的波动性较大。图 7 中的方法展现了应用的负载处于递增、峰值递减时都能较好地拟合。

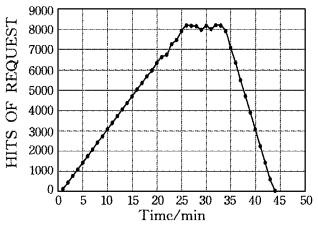

图 4 应用请求量曲线图

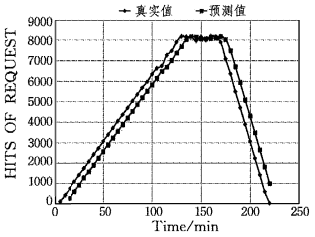

图 5 真实值与 MA 预测方法的预测值曲线图

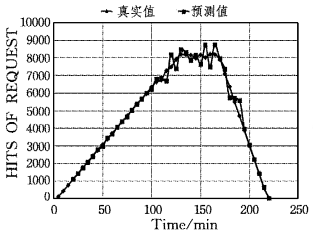

图 6 真实值与 POLYNOM 预测方法的预测值曲线图

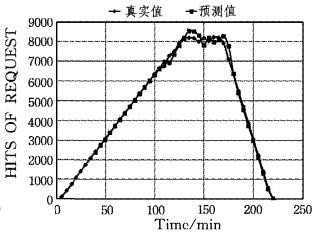

图 7 真实值与 ARIMA 预测方法的预测值曲线图

为了更加精细地观察该场景的真实值与预测值的情况，实验引入了误差进行分析。图 8 给出了 3 种预测方法的相对误差情况，从中也可看出 ARIMA 的相对误差比较平稳，且在 3 种方法中是最小的。在具有高并发、大流量特点的国家科技计划项目申报系统中采用该模型进行预测实验。

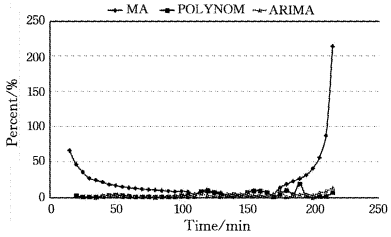

图 8 MA，POLYNOM 和 ARIMA 的相对误差曲线图

6.3 资源调度实验结果和分析

在虚拟机调度算法实验中使用了前述的预测模型，通过预测，在用户访问量持续增大时前述预测模型会触发虚拟资源动态调度算法，进而 VRDS 算法通过计算所需资源的规模来进行资源的合理调度，执行虚拟机的创建或者销毁操作。在默认策略下，实验给予固定的虚拟机数量，在 VRDS 策略下会根据负载的情况进行及时的资源调度。图 9 为两种不同策略下用户访问应用时请求失败数量的对比图，从中可以看出 VRDS 策略能够有效减少用户请求失败的次数。假定在实验中应用要求对每个用户请求响应的最大时间是 2s，图 10 给出了两种策略下应用请求数量的不断递增对用户请求的响应时间，可以看出 VRDS 策略能够比较接近预先设定的响应时间。

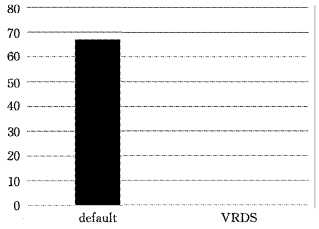

图 9 默认策略与 VRDS 策略下请求失败数量对比图

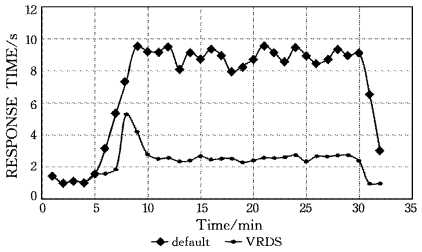

图 10 默认策略与 VRDS 策略的响应时间对比图

图 11 给出了在不同负载的情况下使用默认策略与 VRDS 策略时虚拟机数量的变化情况。从图 11 中可以看出，VRDS 能够在请求超过一定数量时进行虚拟机的创建，当过了高峰期之后，随着负载的不断降低，所需要的虚拟机数量也会不断减少。

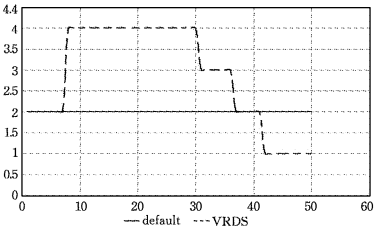

图 11 不同阶段虚拟机数量变化图

综上所述，在前述应用中，本文所采用的预测模型能够较好地拟合应用请求量，不断递增到最大的请求量之后再逐渐递减的场景，VRDS 算法结合本文中的预测模型能够有效、及时地应对应用出现高负载的情况。

结束语 针对不断增长的高并发和用户流量的激增、合理利用资源、保障云应用的稳定运行的问题，本文提出了一种基于预测的虚拟资源动态调度方法，预测有助于应用更早做出应对负载过重的决策，变被动为主动。通过主动计算应对当前负载所需要的虚拟资源的规模以及评估创建虚拟机的合理位置，我们将能更加迅速地应对云应用负载过重的情况，保证应用能够轻松应对海量用户的访问。

当然，实验中将服务器端的负载简化为用户对应用的请求还不足以完整地表达负载的实际状况，负载的预测方法还不够精细，场景也比较简单，我们在未来的工作中将考虑更多的因素，在更加接近实际的环境下进行模拟实验，将更加完善的算法应用到更多的实际场景中。

参 考 文 献

[1] SOMASUNDARAM T S, GOVINDARAJAN K. CLOUDRB: A framework for scheduling and managing High-Performance Computing (HPC) applications in science cloud[J]. Future Generation Computer Systems, 2014, 34(5): 47-65.

(下转第 44 页)

networks?[J]. PLoS Genet,2006,2(6):826-834.
[20] ZOTENKO E,MESTRE J,O'LEARY D P,et al. Why do hubs in the yeast protein interaction network tend to be essential:re-examining the connection between the network topology and essentiality[J]. Plos Computational Biology,2008,4(8):e1000140.
[21] CHUA H N,TEW K L,LI X L,et al. A unified scoring scheme for detecting essential proteins in protein interaction networks [C]//Proceedings of the 2008 20th IEEE International Conference on Tools with Artificial Intelligence(ICTAI'08). USA:IEEE Computer Society,2008:66-73.
[22] LI M,ZHANG H,WANG J X,et al. A new essential protein discovery method based on the integration of protein-protein interaction and gene expression data[J]. BMC Systems Biology,2012,6(1):15.
[23] ZHANG X,XU J,XIAO W X. A new method for the discovery of essential proteins[J]. PloS One,2013,8(3):e58763.
[24] LUO J,MA L. A new integration-centric algorithm of identifying essential proteins based on topology structure of protein-protein interaction network and complex information[J]. Current Bioinformatics,2013,8(3):380-385.
[25] LI M,ZHENG R,ZHANG H,et al. Effective identification of essential proteins based on priori knowledge,network topology and gene expressions[J]. Methods,2014,67(3):325-333.
[26] JIANG Y,WANG Y,PANG W,et al. Essential protein identification based on essential protein-protein interaction prediction by Integrated Edge Weights[J]. Methods,2015,83:51-62.
[27] http://dip.doe-mbi.ucla.edu/dip/Download.cgi?SM=7.
[28] http://www.ncbi.nlm.nih.gov/geo.
[29] TU B P,KUDLICKI A,ROWICKA M,et al. Logic of the Yeast Metabolic Cycle:Temporal Compartmentalization of Cellular Processes[J]. Science,2005,310:1152-1158.
[30] ASHBURNER M,BALL C A,BLAKE J A,et al. Gene ontology:tool for the unification of biology. The Gene Ontology Consortium[J]. Nat. Genet.,2000,25(1):25-29.
[31] YELLABOINA S,TASNEEM A,ZAYKIN D V,et al. DOMINE:a comprehensive collection of known and predicted domain-domain interactions[J]. Nucleic Acids Res.,2011,39(supp_1):730-735.
[32] O'BRIEN K P,REMM M,SPMMHAMMER E L. Inparanoid:a comprehensive database of eukaryotic orthologs[J]. Nucleic Acids Res.,2005,33:D476-480.
[33] MEWES H W,AMID C,ARNOLD R,et al. MIPS:analysis and annotation of proteins from whole genomes[J]. Nucleic Acids Res.,2004,32(Suppl 1):41-44.
[34] CHERRY J M,ADLER C,BALL C,et al. SGD:Saccharomyces genome database[J]. Nucleic Acids Res.,1998,26(1):73-79.
[35] ZHANG R,LIN Y. DEG 5.0,a database of essential genes in both prokaryotes and eukaryotes[J]. Nucleic Acids Res.,2009,37(Suppl. 1):D455-D458.
[36] http://www.sequence.stanford.edu/group.

(上接第 18 页)

[2] ZHAO Y,LI Y,RAICU I,et al. Enabling scalable scientific workflow management in the Cloud[J]. Future Generation Computer Systems,2015,46(c):3-16.
[3] RAMNARAYAN J,MOZAFARI B,WALE S,et al. SnappyData:A Hybrid Transactional Analytical Store Built on Spark [C]//International Conference on Management of Data. ACM,2016:2153-2156.
[4] YUAN H,LI C,DU M. Optimal Virtual Machine Resources Scheduling Based on Improved Particle Swarm Optimization in Cloud Computing[J]. Journal of Software,2014,9(3):705.
[5] HUANG Q,SHUANG K,XU P,et al. Prediction-based Dynamic Resource Scheduling for Virtualized Cloud Systems[J]. Journal of Networks,2014,9(2):375-383.
[6] ACETO G,BOTTA A,DONATO W D,et al. Cloud monitoring:A survey[J]. Computer Networks,2013,57(9):2093-2115.
[7] CS S,S B M S. A Comparative Analysis of Scheduling Policies in Cloud Computing Environment[J]. International Journal of Computer Applications,2013,67(20):16-24.
[8] ZHANG Q,CHEN H,SHEN Y,et al. Optimization of virtual resource management for cloud applications to cope with traffic burst[J]. Future Generation Computer Systems,2016,58:42-55.
[9] ZHENG Q,LI R,LI X,et al. Virtual machine consolidated placement based on multi-objective biogeography-based optimization[J]. Future Generation Computer Systems,2016,54(C):95-122.
[10] LIU Z,ZHOU H,FU S,et al. Algorithm Optimization of Resources Scheduling Based on Cloud Computing[J]. Journal of Multimedia,2014,9(7):1451-1456.
[11] SHAO Y. Virtual Resource Allocation based on Improved Particle Swarm Optimization in Cloud Computing Environment[J]. International Journal of Grid & Distributed Computing,2015,8(1):228-233.
[12] HASSAN M M,ALAMRI A. Virtual Machine Resource Allocation for Multimedia Cloud:A Nash Bargaining Approach[J]. Procedia Computer Science,2014,34:571-576.
[13] SINGH S,CHANA I. Q-aware:Quality of service based cloud resource provisioning[J]. Computers & Electrical Engineering,2015,47:138-160.
[14] SALAH K,ELBADAWI K,BOUTABA R. An Analytical Model for Estimating Cloud Resources of Elastic Services[J]. Journal of Network & Systems Management,2016,24(2):285-308.
[15] SHYAM G K,MANVI S S. Virtual resource prediction in cloud environment:A Bayesian approach[J]. Journal of Network & Computer Applications,2016,65(C):144-154.
[16] HASSANI H,SILVA E S. Forecasting with Big Data:A Review [J]. Annals of Data Science,2015,2(1):5-19.
[17] ALTINTAS N,TRICK M. A data mining approach to forecast behavior[J]. Annals of Operations Research,2014,216(1):3-22.
[18] HANSUN S. A new approach of moving average method in time series analysis[C]//New Media Studies. IEEE,2013:1-4.
[19] https://en.wikipedia.org/wiki/Moving_average.
[20] https://en.wikipedia.org/wiki/Polynomial_regression.
[21] LI J,SHEN L,TONG Y. Prediction of Network Flow Based on Wavelet Analysis and ARIMA Model[C]// International Conference on Wireless Networks and Information Systems. IEEE Computer Society,2009:217-220.
[22] https://en.wikipedia.org/wiki/Autoregressive_integrated_moving_average.