云环境下面向暴发式任务请求的资源部署模型设计

陈 鹏[1] 马自堂[1] 孙 磊[1] 孙冬冬[2]
(解放军信息工程大学三院 郑州 450004)[1] (61579部队 北京 102400)[2]

摘 要 针对云计算环境面临的暴发式任务请求对系统性能带来的影响,提出了一种资源部署模型BWA来应对上述问题。首先由模型的负载监听模块负责监测云计算系统任务请求的变化量,实时判断暴发式任务请求的始末。然后通过引入新的资源部署策略,来避免局部热点的产生,加快系统的响应速度。最后利用跟踪预测算法预置计算节点来进一步加快云计算系统为用户提供服务的速率。通过CloudSim对资源部署模型进行了实验仿真,结果证明,该模型可有效优化系统响应速度。

关键词 云计算,资源部署,暴发式任务请求,负载监听,跟踪预测
中图法分类号 TP302.7 **文献标识码** A

Resource Deployment Model Design in Cloud Computing under Bursty Workloads

CHEN Peng[1] MA Zi-tang[1] SUN Lei[1] SUN Dong-dong[2]
(The Third Institute, PLA Information Engineering University, Zhengzhou 450004, China)[1]
(61579 Army, Beijing 102400, China)[2]

Abstract Aiming at the degrading system performance that bursty workloads bring in cloud computing, BWA(Bursty Workloads Allocation) model was proposed to resolve resource deployment problems. Firstly, BWA's workload monitor model is responsible of monitoring the variation of tasks in cloud computing, judging the on-off in real time about bursty workloads. Next, BWA tries to avoid the appearance of partial hot dots by using new deployment strategy to speed up the system response. At last, the forecasting algorithm is used to improve response speed by deploying the computing nodes in advance. The results of simulation in CloudSim prove that using BWA model can obtain better system performance.

Keywords Cloud computing, Resource allocation, Bursty workloads, Monitor workloads, Forecast workloads

1 引言

对于许多企业的首席信息官(Chief Information Officer, CIO)来说,之所以选择云计算来代替企业自身去架设服务器,是因为云计算模式不仅节省了大量服务器初期的建设成本;还节约了大量后期的运维成本,更重要的一点是,对于某些服务密集型的企业,用户的需求往往波动剧烈,而且在特定时间或特定事件的驱动下会呈现出类似于波峰波谷式变化特征的任务请求量[1]。在云计算出现之前,为了不降低用户的QoS体验、满足与用户签订的SLA,企业往往选择建立足够的计算能力以满足波峰的需求,但随之而来的就是大量冗余的计算能力所造成的巨额的成本增加[2]。

云计算这个看似拥有"无限"计算能力的资源池的出现使得上述问题迎刃而解[3],但随着各类型、不同规模的用户纷纷加入云计算,后者也不断面临着运维压力。相对其他问题对云计算应用性能带来的影响,暴发式任务请求对云计算性能的影响尤为严重。暴发式任务请求与其他常见的任务请求相比,主要的异同点是:在相对较短的时间内,存在着海量的并发式服务请求,造成队列堵塞,使云计算提供商无法满足用户的SLA协议。在目前云计算资源调度模型的研究中,对云计算环境下暴发式任务请求的研究很少,主要原因是云计算尚处于发展阶段,用户的参与度远远没有达到预期,并且随着时间的推移、技术的不断成熟和企业的不断推广,以当前的云计算的计算能力应对暴发式任务请求带来的挑战,仍存在着诸多不足。美国Northeastern University的Waleed Meleis等人,对暴发式任务请求进行了研究[4],但只是初步解决了峰值的监测,而没有考虑到云计算自身的规模即对于不同的云计算系统峰值检测值应有不同的标准。另一方面,云计算作为一种服务,其服务对象的行为符合一定周期性的规律,掌握并应对这种规律可以有效地提高资源部署的效率以及云计算系统整体的性能表现,但当前针对暴发式任务请求的资源部署模型中均未作此方面的考虑。

正是基于上述问题,本文提出了一种暴发式任务请求环境下的资源部署模型(Bursty Workloads Allocation, BWA),

到稿日期:2012-11-06 返修日期:2013-03-06 本文受武器装备预研重点基金项目(9140A15060311JB5201)资助。
陈 鹏(1988—),男,硕士生,CCF学生会员,主要研究方向为云计算、系统优化,E-mail:skyskyasd@163.com;马自堂(1962—),男,教授,主要研究方向为信息安全、密码系统工程;孙 磊(1973—),男,博士,副研究员,主要研究方向为云计算基础设施可信增强、可信虚拟化技术;孙冬冬(1975—),女,讲师,主要研究方向为计算机软件应用。

其充分考虑了云计算环境下的暴发式任务请求的影响，同时提炼历史记录对云计算环境下短期满足季节性规律的任务请求量做出预测，以借此达到整体提升云计算系统响应速度的目的。

2 BWA 模型构建

BWA 资源部署模型主要针对云计算的基础设施即服务层(IaaS)[5]，通过对底层资源部署策略的设计来解决暴发式任务请求对系响应速度造成的影响[6]，从而使云计算更加满足其动态可伸缩、绿色计算的特性。

为构建系统架构，首先，不失一般性，本文假设云计算系统当前拥有 m 个用户$\{C_1, C_2, \cdots, C_m\}$和 n 个在线虚拟机(可使用资源点)$\{S_1, S_2, \cdots, S_n\}$($m \geqslant 0, n \geqslant 0$)。BWA 系统架构如图 1 所示。

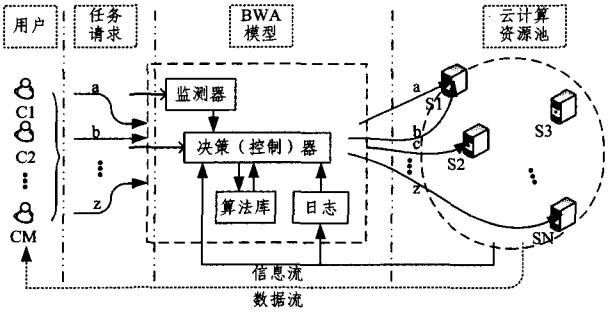


图 1 BWA 系统架构

BWA 模型主要由监测器、算法库、日志、决策(控制)器组成。其中，监测器完成的任务是实时监测用户的任务请求信息，通过内置的算法判断当前的任务是否为暴发式任务请求。算法库用来存储应对不同场景下负载请求的资源分配算法(可由用户添加自己偏好的资源部署算法)，供决策器调用。日志负责记录云计算资源池不同时段的任务请求量信息，为决策器提供实时的横纵向的比较。最后决策器通过综合分析用户、监测器、云计算资源池实时状态和日志所提供的信息，生成资源分配策略，完成资源部署。本文在总结现有的资源分配策略的基础上，增加了对云计算中暴发式任务请求因素的考虑，设计了负载监测器、日志管理等一系列的应对手段，其具体的工作流程如图 2 所示。

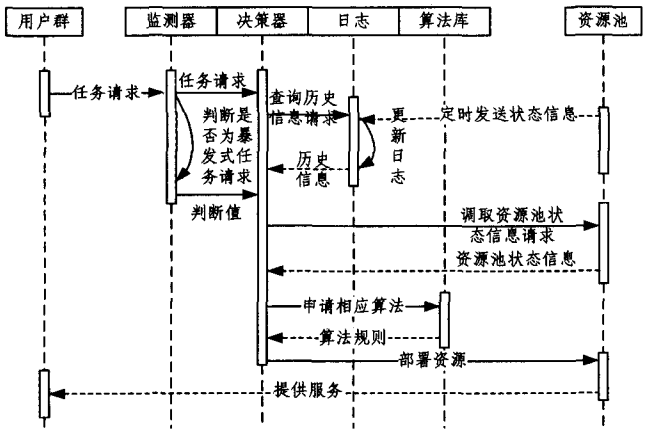


图 2 BWA 资源部署模型工作时序图

工作流程可论述为：

1)任务请求监测：监测器负责实时地接收用户的负载请求信息，通过对请求量变化率与变化量绝对值的综合分析来判断当前请求是否为暴发式任务请求，并将判断结果传递给决策器。

2)综合分析：决策器调用日志、决策器、资源池的即时信息，对负载监听指数与跟踪预测算法所提供的数据进行综合分析，为指定部署节点提前预置计算节点提供依据。

3)决策：决策(控制)器通过前一阶段的分析值选择适当的算法完成任务需求，并将相应的任务请求指定到目标服务器上。

4)资源部署：云计算资源池端开启相应的服务器设备，部署相应的虚拟机(计算节点)来完成整个流程的部署，为用户提供服务。

3 监测器负载监听指数及算法设计

3.1 负载监听指数设计

监测器为 BWA 模型的特征部件，是实现云计算环境下暴发式任务请求的核心，为判断当前负载强度是否达到暴发式级别提供了直观参数。

本文在这里引用暴发式负载监听指数 I 作为标识负载强度的变量[7]。其表达式如下：

$$I = SCV\left(1 + 2\sum_{k=1}^{\infty}\rho_k\right) \tag{1}$$

其中变量的平方系数(Squared Coefficient of Variation，SCV)为一个固定长度任务请求量的平方系数，ρ_k 代表自相关系数，是一种用来寻找随机变量与系统自身关系的统计学方法。假设一个时序的系统随机变量值为$\{X_n\}$，其中 $n=(0,1,\cdots,\infty)$，则：

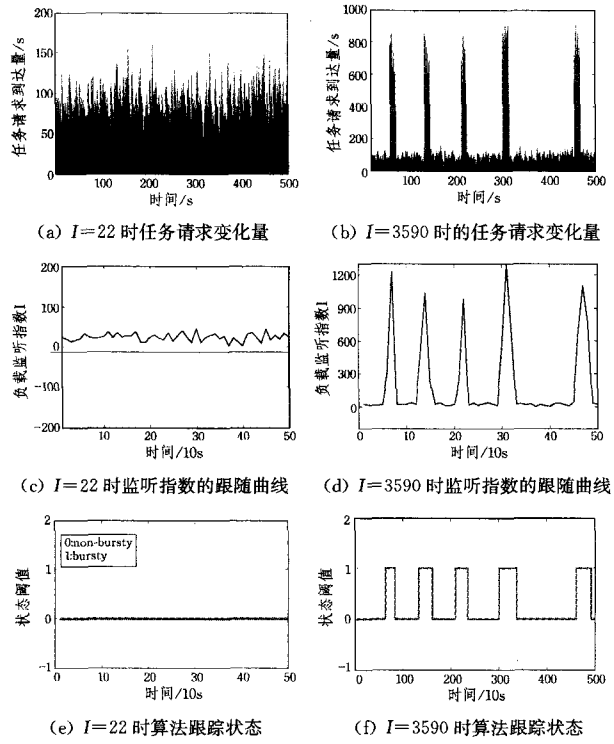
$$\rho_k = \frac{E[(X_t - \mu^{-1})(X_{t+k} - \mu^{-1})]}{\sigma^2} \tag{2}$$

式中，μ^{-1}代表均值，σ^2 为变量$\{X_n\}$的方差。式(1)中 k 的取值范围是(0～∞)，但是很明显这对于实际计算 I 值带来了麻烦。本文要求设定一个固定的请求长度 K 作为计算 I 的长度节点，其中 $K = arrival\_rate \times C\_requests$，$C=1000$，这里 C 为设定值，本文取 1000 主要是因为对于监测暴发式任务请求模型，不但要及时把握任务请求量的最新变化，而且还要提取足够的样本数量以提供必要的信息量来对任务请求做出准确的预测，因此本文选取 1000 个样本作为一个预测周期。

为具体说明负载监听指数 I 作为一种简易的单一变量是如何快捷捕捉暴发式请求信息的，本文首先选取 10s 作为一个时间窗口(整个实验时间长度为 500s)，并设计两种任务请求强度，如图 3(a)，(b)所示。图 3(a)中(SCV＝20)有 $I \approx SCV=20$，这主要是由于负载的分布趋向于标准正态分布，即 $\sum_{k=1}^{\infty}\rho_k \approx 0$。由此可以看出，负载监听指数 I 可以有效地监测到暴发式任务请求而不受小范围内任务请求变化量的干扰。图 3(b)中，$I=3590$，可较好地反映出暴发式负载。从图 3(c)可以很好地看出，虽然(a)情况下，系统的负载请求量在不断地变化，但 I 值始终在一个小的范围内变动，没有误判暴发式任务请求的现象发生。而在(d)图中，从 y 轴数值变化量上可以看出 I 值紧紧地跟随着任务请求的变化，且 I 值变化幅度的绝度值也相对较大。

目前类似的暴发式任务请求现象已经广泛出现，例如铁道部的订票网站上，每到暑假或是春节等特殊时段，铁道部官网上经常发生网站崩溃或长时间延迟的现象，其中很大一部分原因是由于短时间内大量集中的用户任务请求所造成的。

云计算系统计算能力虽然强大，但随着用户的不断加入、业务量的不断激增，长远来看，也面临着此方面的压力。



(a) I=22 时任务请求变化量　(b) I=3590 时的任务请求变化量

(c) I=22 时监听指数的跟随曲线　(d) I=3590 时监听指数的跟随曲线

(e) I=22 时算法跟踪状态　(f) I=3590 时算法跟踪状态

图 3　不同任务请求强度下负载监听指数对比

3.2　监测器算法设计

在计算出暴发式负载监听指数 I 后，接下来的任务就是根据暴发式负载监听指数 I 的值确定何种情形下的任务请求可被定义为暴发式任务请求。与此同时，算法也要能够清晰地判明暴发式任务请求何时开始，何时结束。本文提出的监测器算法如表 1 所列，首先计算当前批量请求与上一时刻批量请求差值的绝对值，并与 n_1($n_1>1$)倍的 SCV 值做比较。此做法的目的是排除普通任务请求变化的干扰(即非暴发式任务请求)，n_1 的取值需根据云计算资源池所能提供的计算能力自行设定。如果等式成立，则比较前后 I 值的变化率，如果大于 n_2 或小于$\frac{1}{n_2}$，就可以初步判明当前负载为暴发式负载，不选用相同的 n 值主要是因为这样不仅可以从绝对值上来度量负载的变化情况，还可以从负载变化的幅度上进行判断，从而可以有效避免单一的判断指标造成误判现象的发生。接下来应判断其为暴发式负载的开端还是结束，比较 R_c 与 R_t 的值，当 $R_c>R_t$ 时，即当前批量请求的到达率大于整个系统的平均到达率，则可以认为当前的负载请求为暴发式的，反之亦然。最后，如果算法断定本次批量的任务请求量不满足暴发式负载请求的条件时，系统将 I_c 的值赋给 I_o，为进行下一次判断做准备。

图 3(e)、(f)为应用 Estimate bursts' algorithm 后系统的状态值。从图 3(e)中可知，算法成功地排除了普通任务请求变化的干扰，没有误判成暴发式负载请求。可以看到图 3(f)中，状态的变化量很好地跟随了暴发式负载的开始与结束，但存在微小的滞后现象，这主要是由系统提取批量数据(k)的等待时间和系统处理信息与通讯的延时造成的，但对于用户来说，这些微量的延时可以忽略不计[8]。

表 1　Estimate bursts' algorithm

Input
K, a batch of request(K>0).
X_n, the nth numbers of request. (X_n>0).
SCV, the squared-coefficient of variation.
n_1, the index of dispersion.
n_2, the index of ratio.
Estimate the bursts
1. Ic←calculate I for current batch.
2. Rc←the arrival rate for current batch
3. Rt←the arrival rate for total batch.
4. SCV←users defined.
5. if($|I_c-I_o|>n_1$ * SCV)then //Io: the old I for current batch;
6. { if($\frac{I_o}{I_c}>n_2 \cup \frac{I_o}{I_c}<\frac{1}{n_2}$)
7. {for(R_c, R_t; $R_c>R_t$; "burst stsrts", goto 13)
8. else
9. "burst ends", goto 13}
10. end if}
11. $I_o=I_c$, "no burst"
12. end if
13. end

4　资源部署策略

在拥有判断暴发式任务请求的能力以后，接下来的任务就是设计一种可有效应对云计算环境下暴发式任务请求的资源部署策略，以给用户带来流畅、满意的体验。

4.1　暴发式负载放置策略

目前，对云计算虚拟机部署算法方面的研究已初步形成，并可以将常见的部署算法归纳为"贪婪式"算法("贪婪式"算法特征：总是遵循既定的算法选择性能最优的节点作为部署目标)和与之相对应的"随机式"算法("随机式"算法：随机地选择计算节点作为部署目标)。当前虚拟机的部署策略综合考虑负载均衡与服务器性能指标等因素，但其在暴发式任务请求的环境下，"贪婪式"算法面对的挑战为算法产生的延时判断响应时间在应对暴发式任务请求时所造成的系统资源部署策略失效[9]，进而违背了与用户签订的 SLA，降低了用户的使用体验。究其原因，主要是当系统面临用户暴发式任务请求时，当前的资源部署策略仍然通过实时调取系统资源使用情况，选择出最优的(不同算法针对不同的性能：从负载均衡角度出发，或从性能最优化角度出发，或从部署时间角度出发等等)目标物理机来对资源进行部署。但其均需要消耗少量的运算时间与通讯时间，在常规负载请求强度的情形下，并不会对系统的响应时间造成可感知的影响，但是在暴发式任务请求情形下，由于任务请求到达很集中，系统的负载均衡、性能占用等情况变化十分剧烈，在应用传统部署算法时，少量的系统滞后时间就会对系统负载的真实情况造成误导，使得系统的资源部署策略依然遵循前一时刻的部署位置，造成大量局部热点的产生，降低系统整体的性能表现。

基于上述问题，本文提出一种先粗后细变化控制粒度的资源部署策略来应对暴发式任务请求，其具体思想描述为：

1)当监测到暴发式负载到达时，使资源部署策略在限定范围内偏向"随机"式算法，即尽量避免因追求某些特定指标的最优解而造成的整体性能的下降；

2)当监测结果显示为 non burst 或是暴发式负载结束时，资源部署策略不变或应及时地转换为偏向"贪婪式"算法，即在不影响系统整体性能的前提下，追求性能表现的最佳。

下面本文统一将追求某方面特定性能的算法统称为

"greedy"算法，而将随机性的分配算法统称为"random"算法。两种部署算法的本质区别在于对候选节点数量的把握，因此，本文结合两大类算法的优点，并尽可能地避免双方的短板，提出了 BWA 云计算环境下暴发式任务请求资源分配算法。其核心思想如表 2 所列。

表 2　the high level of the BWA

input
N, the number of available site.
K, the candidate sites(1<K<N).
I, the state of system(burst or non burst).
The algorithm of BWA
1. if(detect the start of burst)
2. {set K to Nu_b;}// Nu_b is close to N; such as Nu_b=1/2N.
3. set k to Nu_s;// Nu_s to be a small value; such as 1
4. end if
5. analysis all sites S_i;//1<i<N
6. select S={S_1, S_2, …, S_k}//select out the best K sites.
7. select S′=uniform(1, K)//under the random measure.
8. submit the job to S′.

在暴发式任务请求状态下，Nu_b 的取值对于系统的性能表现起到至关重要的作用，而且需要自行调试适合不同云计算系统的取值。在下节的实验部分，本文在固定任务请求量（即 I 值相同）且可用资源节点数一定的情况下，通过使 Nu_b 选取不同的值反复比较响应速度，最终选取 $Nu_b=1/2N$ 为本文实验环境的最佳取值。

4.2　跟踪预测算法

虽然云计算系统中的任务请求强度无法事先预知，但利用暴发式负载监听指数 I 可以很好地判断暴发式负载的始末。同样，云计算作为一种商业计算模式，其用户的行为同样遵循某种特定的规律，以年为度量单位时间，云计算的业务量势必呈现递增态势，而以周为一个时间度量单位，云计算的业务量又呈现出一种类似正弦函数式的波动分布，即在一个小范围内按照一定的规律反复波动。在云计算系统部署资源的过程中，从选定目标主机到将虚拟机部署在其上实质上是一种对用户需求的滞后响应，会对整个系统性能造成一定的影响，因此提前预测并预制好资源是有效应对此类问题可行的解决方案。

利用统计学方法 Holt-Winters[10] 季节指数平滑模型可对具有时效性的且符合一定规律的任务请求活动进行建模。本文根据云计算的特点选用加法模型[11]，首先给出模型输入变量 x_t 的平滑指数 $\hat{x}_t$ 表达式：

$$\hat{x}_{t+k}=m_t+n_tk+S_{t+k},\ t=s,s+1,\cdots,T$$

式中，m 代表截距即当前任务请求量；n 代表斜率即任务请求量的变化率；S 为系统的季节因子；s 为周期的长度，本文选取 s=48 即选取周为一个时间周期长度。下式为平滑指数表达式中 3 个参数的递推公式：

$$\begin{cases} m_t=\alpha(x_t-S_{t-s})+(1-\alpha)(m_{t-1}+n_{t-1}) \\ n_t=\beta(m_t-m_{t-1})+(1-\beta)n_{t-1} \\ S_t=\gamma\dfrac{x_t}{m_t}+(1-\gamma)S_{t-s} \\ 0<\alpha,\beta,\gamma<1 \end{cases}$$

最后得出预测值：

$$\hat{x}_{T+k}=m_T+n_Tk+S_{T+k-s}$$

（S_{T+k-s} 为样本数据最后一年的季节因子），接下来给出模型的初始值计算公式。

a）计算前两个周期的平均任务请求增量

$$\overline{V_1}=\frac{1}{S}\sum_{t=1}^{S}x_t$$

$$\overline{V_2}=\frac{1}{S}\sum_{t=S+1}^{2S}x_t$$

b）计算初始截距值与初始斜率

$$n_{2S+1}=\frac{1}{S}(\overline{V_2}-\overline{V_1})$$

$$m_{2s+1}=\overline{V_2}+\frac{S-1}{2}n_{2s+1}$$

c）计算前两个周期的季节因子

$$S_{1t}=x_t-\overline{V_1}+(\frac{S+1}{2}-k)n_{2S+1}$$

$$S_{2t}=x_{t+S}-\overline{V_2}+(\frac{S+1}{2}-k)n_{2S+1}$$

d）计算前两个周期的平均季节因子

$$S_t''=\frac{1}{2}(S_{1t}+S_{2t})$$

e）将季节因子做正态化处理

$$S'=\sum_{t=1}^{S}S_t'',\ S_t=\frac{S}{S'}S''$$

Holt-Winters 季节指数平滑模型可以很好地对具有季节性规律的时间序列进行准确的预测，为云计算系统提前部署资源以提高系统整体的响应速度提供理论依据。

5　分析与实验

本文采用云计算仿真软件 CloudSim 作为仿真平台[12]，利用 Matlab 矩阵实验室模拟生成任务请求数据量，用以模拟现实世界云计算系统与用户间的交互频次。

1）实验环境声明

实验采用 Windows XP SP3 操作系统；云仿真平台选用 CloudSim2. 1. 1 版本；Matlab2008a；JDK 版本为 jdk1. 6. 0_10。

2）CloudSim 扩展编译

通过对 CloudSim 基础类进行扩展编译实现本文提出的 BWA 模型。

a）UtilizationModel

在接口 UtilizationModel 添加负载监听指数判断方法。

b）VmAllocationPolicy

在 VmAllocationPolicy 类中添加 VmAllocationolicy_greedy 继承类、VmAllocationPolicy_random 继承类、VmAllocationPolicy_BWA 继承类，其分别对应"贪婪式"、"随机式"和本文提出的 BWA 模型中的算法，为后续的性能对比实验做铺垫。

c）CloudCoordinator

在 CloudCoordinator 类中添加 updateNote 方法，通过定时调用 updateDatacenter 方法中的参数完成 updateNote 方法中对跟踪预测算法的实现。

3）实验仿真

本文主要以系统响应时间 t 作为参考依据，分别模拟暴发式任务请求环境下与非暴发式任务请求环境下 3 种类型算法的响应时间，并通过对比得出结论。实施步骤如下：

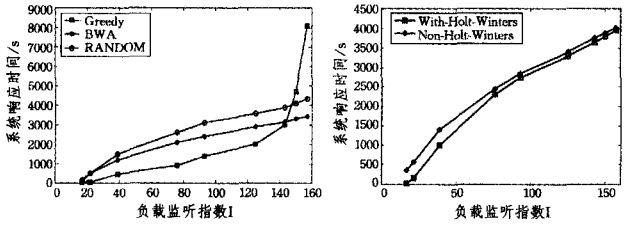
a）在所编写的仿真程序中，设定一个拥有 100 台服务器的数据中心，并假定有 300 台可用的虚拟机节点。生成的随机数用以模拟单位时间内并发的任务请求数量（假定每个任

务的长度相同),统计在不同 I 值情况下系统的响应时间。

b)以 a)中生成的任务请求数为依据,以周为一个时间周期,利用本文提出的 Holt-Winters 季节指数平滑模型对下一周的任务请求数做出预测,并提前为每天预先部署相应数量的虚拟机。记录并统计不同任务请求强度下系统的响应时间。

4)实验结论

实验结果如图 4 所示,在图 4(a)中,对 3 种算法在不同负载监听指数强度下的系统响应时间取 9 个点进行描述。本文可以得出结论,即在没有暴发式任务请求的环境下,Greedy 算法有着更佳的表现,但是随着系统瞬时任务请求量的不断增加,本文提出的 BWA 模型可有效地降低暴发式负载对于系统响应时间的影响。通过观察图 4(b)本文可以得出以下结论:Holt-Winters 模型可有效预测系统下一个时间周期的任务请求数量,在低负载率(非暴发式任务请求)的环境下,可更好地加快云计算系统的资源部署过程。



(a) 3 种算法在不同负载监听指数强度下的系统响应时间

(b) 添加跟踪预测算法前后系统响应时间比较

图 4 BWA 模型对系统响应速率的影响

结束语 本文针对云计算系统会遇到瞬时大量任务请求的问题,设计了 BWA 模型以应对暴发式的任务请求。其中主要的工作从以下两点展开:1)设计了负载监听指数 I,用以实时监控云计算系统任务请求变化量,并通过监测器算法动态地判断暴发式任务请求的始末;2)增加了 Holt-Winters 季节指数平滑模型,以预测云计算系统小规模的符合季节性变化规律的任务请求量,从而加快云计算系统的响应速度。最后通过实验仿真证明本文提出的模型可以有效地提高云计算系统在暴发式任务请求量下的响应速度,提高用户体验。

参 考 文 献

[1] 谭一鸣,曾国荪,王伟. 随机任务在云计算平台能耗的优化管理方法[J]. 软件学报,2012,23(2):266-278

[2] Cui V,Zhou T,Chen J,et al. 中国云计算发展之道 [R]. 北京:IDC 中国,2010

[3] 杨星,马自堂,孙磊. 云环境下基于改进蚁群算法的虚拟机批量部署研究[J]. 计算机科学,2012,39(9):33-37

[4] Tai Jiang-zhe,Meleis W,Zhang Jue-min,et al. ARA:Adaptive Resource Allocation for Cloud Computing Environments under Bursty Workloads[R]. 978-1-4673. Northeastern University,Boston,USA,2011

[5] 罗军舟,金嘉晖,宋爱波,等. 云计算:体系架构与关键技术[J]. 通信学报,2009,32(7):1337-1348

[6] 杨际祥,谭国真,王荣生. 并行与分布式计算动态负载均衡策略综述[J]. 电子学报,2010,38(5):1122-1130

[7] Gusella R. Characterizing the Variability of Arrival Processes with Indices of Dispersion[R]. TR-90-051. International Computer Science Institute,USA,1990

[8] Fang Mi-ning, Giuliano C, Cherkasova L, et al. Burstiness in Multi-Tier Applications: Symptoms, Causes, and New Models, CNS-0720699[R]. CCF-0811417. College of William and Mary, Williamsburg,USA,2008

[9] Lassnig M,Fahringer T. Identification,modeling and prediction of non-periodic bursts in workloads[C]// 2010 10th IEEE/ACM International Conference on Cluster,Cloud and Grid Computing. Austria,IEEE DOI /CCGRID,2010:485-494

[10] Goodwin P. The Holt-Winters Approach to Exponential Smoothing:50 years Old and Going Strong[J]. FORESIGHT,2010,Fall:30-33

[11] Prajakta S K. Time series Forecasting using Holt-Winters Exponential Smoothing[D]. Kanwal Rekhi School of Information Technology,2004

[12] Rodrigo N C,Ranjan R,Beloglazov A,et al. CloudSim:A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms[R]. Cloud Computing and Distributed Systems (CLOUDS) Laboratory,Australia,2010

(上接第 67 页)

性,采用 AR 模型优化内存算法,减少内存页面的传递数量,缩短虚拟机迁移时间,降低迁移时的网络带宽开销,保证了同台服务器上其它虚拟机的网络带宽应用,提高了云计算环境下虚拟机的性能,优化了云计算系统。如果考虑 AR 模型的普遍性,有些场景是不能胜任的,比如低内存服务环境下的云计算,因为 AR 模型的计算时间反而提高迁移时间。下一步工作就是研究低内存场景。

参 考 文 献

[1] 文明波,丁治明. 适用于云计算的面向查询数据库数据分布策略[J]. 计算机科学,2010,37(9)

[2] Li Bo,Li Jian-xin,Huai Jin-peng,et al. Enacloud:An energy-saving application live placement approach for cloud computing environments[C]//Proceedings of the International Conference on Cloud Computing. Bangalore,2009:17-24

[3] 陈廷伟,周山杰,秦明达. 面向云计算的任务分类方法[J]. 计算机应用,2012,32(10):2719-2723,2727

[4] Menon A,Santos R J,Turner Y,et al. Diagnosing Performance Overheads in the Xen Virtual Machine Environment[C]// Proc of the 1st ACM/USENIX International Conference on Virtual Execution Environments. Chicago,USA:[s. n.],2005:13-23

[5] 刘伯成,陈庆奎. 云计算中的集群资源模糊聚类划分模型[J]. 计算机科学,2011,34(12)

[6] Xen [EB/OL]. http://xen. org

[7] 李强,郝沁汾,肖利民,等. 云计算中虚拟机放置的自适应管理与多目标优化[J]. 计算机学报,2011,34(12)

[8] 张伟哲,张宏莉,张迪,等. 云计算平台中多虚拟机内存协同优化策略研究[J]. 计算机学报,2011,34(12)

[9] Han H,Jung H,Kang S,et al. Performance evaluation of a remote memory system with commodity hardware for large-memory data processing[J]. Cluster Computing,2011,14(4):325-344

[10] Yan Li-ren,Huang Wei. An IC yield enhancement approach by ARMA modeling and dynamic process control[J]. The International Journal of Advanced Manufacturing Technology,2009,42(7/8):749-756

[11] 张文杰,钱德沛,栾钟治,等. bing 算法估测网络带宽的研究与实现[J]. 小型微型计算机系统,2004,25(5)