软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2013,24(8):1937−1946 [doi: 10.3724/SP.J.1001.2013.04364]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

云计算虚拟资源的熵优化和动态加权评估模型*

左利云[1,2], 曹志波[2], 董守斌[2]

[1](广东省石化装备故障诊断重点实验室(广东石油化工学院),广东 茂名 525000)

[2](广东省计算机网络重点实验室(华南理工大学),广东 广州 510641)

通讯作者: 董守斌, E-mail: sbdong@scut.edu.cn

摘 要: 云资源的动态变化和不确定性给资源管理及任务调度带来了很大的困难.为了准确地掌握资源动态负载和可用能力信息,提出一种基于熵优化和动态加权的资源评估模型,其中,熵优化模型利用最大熵和熵增原理的目标函数及约束条件,筛选出满足用户 QoS 和系统最大化的资源,实现最优调度,保障用户 QoS.对筛选后的资源再进行动态加权负载评估,对负载过重及长期不可用资源进行迁移、释放等,可减少能耗,实现负载均衡和提高系统利用率.设计了仿真实验,以验证所提评估模型的性能.实验结果表明,熵优化模型对用户 QoS 和系统最大化有很好的效果,动态加权负载评估有利于均衡负载,提高系统利用率.该评估模型实现了用户 QoS 保障、减少能耗、负载均衡以及提高系统利用率等多目标的优化.

关键词: 云计算;虚拟资源评估;熵;动态负载;多目标优化

中图法分类号: TP316 **文献标识码**: A

中文引用格式: 左利云,曹志波,董守斌.云计算虚拟资源的熵优化和动态加权评估模型.软件学报,2013,24(8):1937−1946. http://www.jos.org.cn/1000-9825/4364.htm

英文引用格式: Zuo LY, Cao ZB, Dong SB. Virtual resource evaluation model based on entropy optimized and dynamic weighted in cloud computing. Ruan Jian Xue Bao/Journal of Software, 2013,24(8):1937−1946 (in Chinese). http://www.jos.org.cn/1000-9825/4364.htm

Virtual Resource Evaluation Model Based on Entropy Optimized and Dynamic Weighted in Cloud Computing

ZUO Li-Yun[1,2], CAO Zhi-Bo[2], DONG Shou-Bin[2]

[1](Guangdong Province Key Laboratory of Petrochemical Equipment Fault Diagnosis (Guangdong University of Petrochemical Technology), Maoming 525000, China)

[2](Guangdong Province Key Laboratory of Computer Network (South China University of Technology), Guangzhou 510641, China)

Corresponding author: DONG Shou-Bin, E-mail: sbdong@scut.edu.cn

Abstract: The dynamic and uncertainty of cloud resource makes resource allocation and task scheduling more difficult. In order to retrieve accurate resource information about dynamic loads and available capacity, this study proposes a resource evaluation model based on entropy optimization and dynamic weighting. The entropy optimization filters the resources that satisfy user QoS and system maximization by goal function and constraints of maximum entropy and the entropy increase principle, which achieves optimal scheduling and satisfied user QoS. Then the evaluation model evaluates the load of having filtered resources by dynamic weighted algorithm. In order to reduce energy consumption, achieve load balancing and improve system utilization, the study allows the migration or release the resources which overload and unavailable for a long time. Experimental results show the effect of entropy optimization on user QoS and system maximization, and dynamic weighted algorithm benefits load balancing and system utilization. The experimental results prove that

* 基金项目: 国家自然科学基金(61070092)

收稿时间: 2012-06-19; 定稿时间: 2012-12-27

the evaluation model achieves multi-objective optimization such as satisfying user QOS, reducing energy assumption, balancing load, improving system utilization and so on.

Key words: cloud computing; virtual resource evaluation; entropy; dynamic load; multi-objective optimization

云资源的动态变化和不确定性给资源分配及任务调度带来了很大的困难,而且云计算环境的复杂性使得传统的资源管理无法满足其多目标需求,因此,如何解决云资源的动态变化问题,实现多目标优化,是目前云资源管理面临的重要问题.

云计算的商业化模式,使得服务质量(即用户 QoS)成为判定服务提供者是否成功的一个重要因素,也成为资源管理和任务调度的首要目标.大量研究者对实际云计算及网格资源可用情况进行追踪与监控,分析资源可用度对应用性能的影响[1−3],结果表明,若资源保持较高的不可用频率,则将对任务的执行产生较大的负面影响,严重影响应用性能和用户 QoS 保障(如完成时间).云资源工作负载的动态性导致资源可用能力存在较大的不确定性,因此,准确地描述云资源可用能力的动态特征、掌握资源动态可用能力信息,对系统选择最佳的资源-任务调度起着直接的指导作用,是实现多目标调度优化的一个关键条件.

在云资源中,虚拟资源占有非常大的比重,虚拟资源具有服务迁移和部署灵活的特点[4],即虚拟机平台能够动态地将服务映射到所需要的物理资源,而正在运行的应用服务程序不必停止或重启.因此,对虚拟资源的可用性及动态负载进行评估,可将不可用频率较高或负载很轻的虚拟机进行动态迁移,关闭闲置物理设施,以减少消耗[5];同时,也可对负载较重的虚拟机通过进程的动态迁移,动态地调整到适合的物理节点,从而实现系统负载均衡,提高系统整体性能.

因此,云资源动态可用能力的评估对资源分配管理、任务调度具有重要意义,对实现负载均衡、减少能耗、保障用户 QoS 和提高系统利用率起关键作用.本文研究工作以此展开.

本文的主要贡献包括:

(1) 提出了一种熵优化模型,筛选满足用户 QoS 和系统最大化的资源;

(2) 提出了一种动态加权负载评估算法,实现负载均衡并减少能耗.

1 相关工作

目前,分布式系统和网格环境中有一些资源评估预测系统如 NWS[6],MDS[7],但它们仅仅是一些简单的评估工作,以资源的物理性能指标评价其服务能力,利用相关预测技术估计资源未来状态.而且研究主要集中在资源的物理性能预测方面,根据对资源的相关物理性能属性(如 CPU 利用率、内存容量、网络带宽等)进行监控得到的历史数据,对未来一段时间内的资源物理性能参数值进行预测[8−12].在一定程度上,对用户 QoS 起到保障作用;但对于云计算这样的动态复杂环境,这些评估指标难以适应云计算的动态变化,无法满足云计算多目标优化的需求.

而在云计算中,还没有比较成熟的资源可用性评估系统.相关研究有:文献[13]提出了一个轻量级的资源管理模型——EAC,提供轻量级的资源管理业务;文献[14]提出了一种评估云资源可信度的框架;文献[15]提出了一种资源供应框架,针对作业需求评估用户应该从服务提供商租用多少资源;文献[16]提出一种虚拟环境下面向服务的多层次监测框架,目的是收集和监测统计网络应用中可能遇到的一些功能限制信息,同时也提供了一个动态自适应的重新配置功能,以求能够有效地适应资源配置的动态质量服务(QoS)等应用的需求;文献[17]提出一种利用轻重信号处理和统计学习算法来实现网上的动态应用程序资源需求的预测.但它们并没有针对资源的可用情况进行评估.

与本文类似的研究是文献[18].它考虑了网格环境的动态变化,对资源可用情况进行研究.它采用求用户和资源供应商选择资源目标函数极值的方法,根据极值函数曲线求出使系统最优的资源候选集,来保障用户 QoS,但它所采用的微分计算方法复杂度较高,因此我们希望提出一种更为简单、有效的动态评估方法,并同时实现负载均衡、减少能耗、保障用户 QoS 和提高系统利用率等多目标的优化.

2 基于熵优化原理的云资源动态可用评估模型

资源可用性是指作为云服务提供者,在一段时间内持续提供云服务的概率和服务能力的大小.资源可用性评估是基于资源可用信息的监控,提取一些准确刻画动态资源可用度和可用能力指标来评估资源的历史行为,以此预测资源的未来状态,为任务调度或资源负载评估提供相对准确的资源实时性能信息,可集成到各种调度系统或资源管理系统,为调度或虚拟资源的迁移提供依据.

2.1 资源可用评估指标

当前,评估指标大多是基于静态或预测的物理性能指标,如 CPU 计算能力、存储容量、网络带宽等,但在云计算这样的动态环境中,这些指标存在不确定性和非标志性(即可能出现两个 CPU 计算能力在数值上相等,但实际处理速度却不同的情况),因此,这些静态指标很难反映一个资源的实际服务能力.

在此,为了描述资源动态负载状态和实际可用能力,采用如下 3 个动态指标参数[18],并假设虚拟资源集合 U 中共包含 n 个资源,即 $U=\{U_1,U_2,...,U_n\}$:

(1) 资源请求量 r:单位时间内资源接收到的平均服务请求数目.若资源节点 $U_i(1\leqslant i\leqslant n)$的资源请求量为 $r_i(1\leqslant i\leqslant n)$,则 U 的资源请求量为 $r_N=\sum_{i=1}^{n}r_i$.

(2) 资源服务能力 h:单位时间内资源完成的平均服务请求数目.h 值越大,表明资源服务能力越强,则资源价格越高,即资源价格与 h 成正比.若资源节点 $U_i(1\leqslant i\leqslant n)$的资源请求量为 $h_i(1\leqslant i\leqslant n)$,则 U 的资源能力为 $h_N=n\Big/\sum_{i=1}^{n}\frac{1}{h_i}$.

(3) 资源服务强度 q:完成一个服务请求的平均时间与服务请求的平均时间间隔的比值,且 $q=\frac{r}{C\cdot h}$,其中,C 为资源的并行服务能力.若资源节点 $U_i(1\leqslant i\leqslant n)$的并行服务能力为 $C_i(1\leqslant i\leqslant n)$,则资源 U 并行服务能力 $C=C_1+C_2+\ldots+C_n$.

资源服务强度体现了资源负载压力情况与其可用能力的相对关系.资源服务强度趋近于 1,表明资源的工作压力趋向满负荷.U 的资源强度为 $q_N=r_N\Big/\left(h_N\cdot\sum_{i=1}^{n}C_i\right)$.

2.2 基于熵优化原理的资源评估模型

采用监控组件监控、跟踪并统计资源的动态信息(如平均服务请求数目、资源平均服务时间),通过曲线拟合技术建立资源请求量与资源服务能力的概率分布模型[19],利用其概率分布模型采用最大熵原理求出资源评估的目标函数,并将资源动态属性作为其约束条件.

关于云计算和网格计算资源评估预测研究[13,14,18,19]中对输入事件(资源请求量 r)的刻画情况显示,输入事件(资源请求量 r)大多呈泊松分布,则其概率分布为

$$P(X=r)=\frac{\lambda^n}{r!}\mathrm{e}^{-\lambda},\ r=0,1,... \tag{1}$$

根据资源请求的概率分布函数,求得其熵值为[20]

$$S=\lambda-\lambda\log\lambda+\sum_{k=0}^{\infty}\log\Gamma(k+1)p(k),\ \lambda>0,\ k=0,1,2,... \tag{2}$$

通过其熵值利用最大熵原理求其最大熵,得出公式(3),并将资源请求量与资源服务能力作为其约束条件,得出公式(6);另根据熵增最小原理,即熵增最小时,系统可达最大化,因此,利用文献[18]中用户选择资源目标函数 $F_u=\alpha\cdot P(r,h)+\beta\cdot W_s(r,h)$与资源提供者选择资源目标函数 $F_p=h\cdot P(r,h)-K\cdot V(r,h)$(其中,$\alpha,\beta$为调节因子,体现用户对费用和截止时间的偏好程度,$P$ 为资源价格,W_s 为服务请求在资源队列中的等待时间与实际服务时间之和,K 为单位时间资源空闲所需支付的成本,$V(r,h)$为资源空闲率[18]),得出公式(7)这一约束条件,再结合其他约束条件求

出当前状态下使系统最优且满足用户 QoS(通过用户选择资源目标函数 F_u 来体现)的资源候选集.在此,将这种最大熵原理和熵增最小原理结合求解的方法称为熵优化模型.

$$\max S_n(P) = -\sum_{i=1}^{n} P_i \ln P_i \tag{3}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} P_i = 1 \tag{4}$$

$$P_i>0,\ i=1,\ldots,n \tag{5}$$

$$\sum_{i=1}^{n} P_i q_{Nj}(r_i) = E[q_{Nj}],\ j=1,2,...,m \tag{6}$$

$$|\, d_0F_u + d_0F_p \,| > d_0F_p' + \sum_{k=1}^{n} F_{uk} \tag{7}$$

其中,n 为资源个数,r_i 为资源请求的可能状态值,$q_{Nj}(\cdot)$为资源计算能力函数,F_p' 为资源动态变化部分熵值,$\sum_{k=1}^{n} F_{uk}$ 为用户请求变动相关部分熵值.

具体执行过程如下:

Algorithm 1. Entropy optimized evaluation.

Input: $\langle r_i,h_i,q_i\rangle$. Output: $\langle r_i,h_i,q_i\rangle$.

Get the Distribution of r_i by Fitting Curves;

IF the Distribution of r_i is Poisson THEN

Calculate the Probability of r_i, the Entropy of S as Formula (2);

ELSE

Calculate the Entropy of S by other Distribution of r_i;

ENDIF

Calculate the mean, variance, etc of r_i and q_i, as Formula (4) to Formula (6);

Get Formula (7) include F_u, F_p; //熵增最小时系统最大化,F_u 和 F_p 均衡选择

Calculate the Max of Entropy $S_{\max}$ by Formula (4);

$S_{\max}$ Subject to the Constraint as Formula (4) to Formula (7);

文献[18]采用求 F_u 和 F_p 极值的方法,根据极值函数曲线求出使系统最优的资源候选集,但这种微分方法计算复杂度较高,而本文提出的熵优化原理模型复杂度相对较低.

3 基于熵优化评估模型的资源评估机制

熵优化模型实现了满足系统最大化且用户 QoS 保障的资源筛选,为了实现负载均衡和减少消耗,还需要对云计算中虚拟资源的动态负载情况进行评估.

3.1 云资源动态加权负载评估

在对虚拟资源的负载进行评估时,资源节点周期性地采用动态加权负载算法(dynamic weighted load algorithm)[21]对自身负载状态进行评估,计算出归一化的相对负载值 $L[i]$.$L[i]$必须能够反映出资源之间的性能差异以及潜在的负载强度,为实现负载均衡提供最佳的决策依据,因此,$L[i]$的定义如公式(8)所示.

$$L[i] = \begin{cases} 1, & \text{if } (r_i = R_i \text{ or } h_i \geqslant H_i \text{ or } q_i \geqslant Q_i) \\ w_1\dfrac{r_i}{R_i} + w_2\dfrac{h_i}{H_i} + w_3\dfrac{q_i}{Q_i}, & \text{Others} \end{cases} \tag{8}$$

其中,r_i,R_i 分别表示资源 U_i 的当前资源请求量和最大请求量;h_i,H_i 为资源 U_i 的当前服务能力和最大服务能力;q_i 和 Q_i 为 U_i 当前服务强度和最大服务强度;r_i/R_i,h_i/H_i,q_i/Q_i 分别为 r_i 对 R_i、h_i 对 H_i、q_i 对 Q_i 进行归一化所得的值,

取值范围均为[0,1].

然后,根据 $L[i]$的值采用双阈值λ_1和$\lambda_2(\lambda_1<\lambda_2)$将虚拟资源的负载状态划分为空闲、正常、过载这 3 种状态.λ_1,λ_2计算如下:

$$\lambda_1=Q-\sigma \tag{9}$$

$$\lambda_2=Q+\sigma \tag{10}$$

其中,Q 为系统所有资源负载强度的平均值,σ为系统负载的标准偏差.

$$Q=\frac{1}{m}\sum_{k=1}^{m}Q_k \tag{11}$$

其中,Q_k为资源 U 的服务强度的平均值,M 为系统中所有资源个数.

$$Q_k=\frac{1}{n}\sum_{k=1}^{n}q_k \tag{12}$$

$$\sigma=\sqrt{\frac{1}{M}\sum_{k=1}^{M}(x_k-\bar{x})^2} \tag{13}$$

其中, $x_k=Q_k,\bar{x}=Q$.

$r_i/R_i,h_i/H_i,q_i/Q_i$可通过动态调节加权值 w_j来改变3个资源参数因素对 $L[i]$的影响权重,因此称为动态加权负载评估算法.采用动态变化影响权重,在每一个评估周期,由公式(14)自适应地动态调节加权值 w_j:

$$w_j=w_0+\mu(w_1-w_0) \tag{14}$$

其中,w_0,w_1均为常数,其范围分别是[0,0.5],[0,1],且 $w_0>w_1$;μ是在[0,1]分布的随机数,公式(14)使得 $r_i/R_i,h_i/H_i,q_i/Q_i$的影响权重在$[w_0,w_1]$之间随机变化,并满足公式(15):

$$\sum_{i=1}^{3}w_j=1 \tag{15}$$

执行过程如下:

Algorithm 2. Dynamic weighted evaluation.

Input: R_i, H_i, Q_i. Output: $L[i]$, U_{state}, k.

Calculate $L[i]$ by Formula (8);

Calculate λ_1 and λ_2 by Formula (9) and Formula (10);

IF $r_i=R_i$ or $h_i\geqslant H_i$ or $q_i\geqslant Q_i$ //$L[i]\geqslant\lambda_2$ THEN

    U_{state}=*Overload*;

ELSE IF $L[i]\leqslant\lambda_1$ THEN

    U_{state}=*Idle*; $k=k+1$; //k 为空闲次数,初始值为 0

ELSE

    U_{state}=*Normal*;

ENDIF

Adaptive Dynamic Adjustment of the weighted value w_j by Formula (14)

3.2 基于熵优化模型的云资源多目标优化评估机制

本文提出的云资源评估模型——EOWLEM(entropy optimized and dynamic weighted evaluation model),通过最大熵及其约束条件和熵增最小原理实现系统最大化且满足用户 QoS 需求的资源筛选,又通过动态加权负载评估根据资源负载情况将可用率较低的资源进行迁移、释放,从而减少能源消耗,对过载资源进行动态调整迁移,实现负载均衡.最终实现用户 QoS 保障、减少能耗、负载均衡、提高利用率等多目标优化.具体实现算法伪代码如下:

Algorithm 3. Entropy optimized and dynamic weighted evaluation.

Input: $\langle r_i,h_i,q_i\rangle$. Output: $\langle r_i,h_i,q_i\rangle$.

do
update(); Algorithm 1;
IF Satisfied User QoS and System Maximize THEN
    Algorithm 2;
ELSE
    flag=0; break;
ENDIF
FOR U_{state}=*Overload* or U_{state}=*Idle* THEN
    Migrate();
    IF $k \geq$*Migrate Threshold* THEN
        flag=0; break;
    ELSE
        Algorithm 2;
    ENDIF
ENDFOR
Registered $\langle r_i,h_i,q_i\rangle$ to Assessment Center;
IF $t \geq$Evaluation Cycle THEN
    Monitor();
ELSE
    flag=1; break;
ENDIF
while $\langle r_i,h_i,q_i\rangle$ Change
IF *flag*=0 THEN
    Migrate(); *Release*();
ELSE
    Output$\langle r_i,h_i,q_i\rangle$
ENDIF

本评估机制首先利用云计算服务监测器来监测统计云资源的一些动态信息,如虚拟机请求数目、虚拟资源的计算时间等,根据这些动态信息,采用曲线拟合技术设计虚拟资源请求数目和虚拟资源计算能力的概率分布模型[20];并及时监控这些信息的变化规律和特征,如工作日与周末的不同等,从而尽可能准确地表示资源的动态可用能力的特征和动态负载情况.用动态加权负载评估算法对熵优化筛选后的资源负载情况进行标识,对虚拟资源的负载情况划分为空闲、正常、过载这 3 个级别.经过以上对虚拟资源可用能力及负载情况的评估,将评估结果向评估器注册,并根据监测器的监测结果与评估结果进行比对,若有变化,及时更新评估器中的注册信息.这样,可以得到有关虚拟资源可用能力及动态负载的量化数据,这些量化信息对系统选择最佳的资源管理分配和任务调度策略起着直接的指导作用.

该方法适用于所有资源的标准可用评估,只需对资源可用情况的概率分布应用熵优化模型及资源动态加权负载评估算法进行评估,即可筛选出适合的资源候选集,并对资源动态负载状态进行评估处理.

4 基于熵优化和动态加权负载评估实验

4.1 实验参数与评价指标

为了验证本文提出的资源评估机制的性能,采用云仿真软件 Cloudsim3.0 进行模拟实验.对资源动态监控方面的研究实现在基于 Linux 系统的平台上.使用 shell 命令 vmstat(虚拟内存统计)可以对系统的 CPU 利用率、

虚拟内存使用情况及进程进行监视,统计系统的整体使用情况;此外,使用 iostat 命令还可以监视磁盘及 I/O 使用情况.资源的整体状态是动态变化的,上述信息需定时统计,为资源的分析评价提供依据.将收集到的资源信息保存在文件 node_infor.txt 中,并定时更新.模拟实验参数设置如下:

(1) 充分模拟云资源动态性,针对不同数目的任务集和节点集进行实验:固定任务数 500,节点数范围 100~1 000,间隔 100;固定节点数为 500,任务数在 100~1 000 范围内,间隔 10.两种情况均选取不同的任务集和节点集进行实验 30 次,取平均值作为实验结果.

(2) 任务属性信息:任务长度[1000,2000],单位 MI;数据传输量[1000,2000];存储量[1000,2000].

(3) 充分模拟云资源数量大、种类多的特征,资源节点属性信息:处理速度[10,200],单位 MIPS;数据存储能力[10000,20000];处理器、负载及网络负载初始值范围均为[0.01,0.1];带宽[100,1000],网络当前延迟为[1,10];节点故障率和网络失效率均在$[10^{-3},10^{-2}]$范围内.

实验中,将本文提出的 EOWLEM 资源评估机制与文献[18]的 AEMJE(availability degree enhanced model for job execution)及以下两种相关技术进行比较:

(1) HA-JES(highly available job execution service)[22]:根据用户对可用度的要求确定执行任务的潜在资源集,将任务调度到资源集中的 1 个或多个资源上,以提高利用率较低的资源的利用率.

(2) ACT(availability check technique)[23]:一种可用性检查技术,用于检查并更新所需资源的状态,当所需资源均为可用状态时开始调度任务.

3 种技术均采用 Min-Min 调度算法来调度任务,每种比较均采用 30 次的运行结果作为最终实验结果.其性能评价指标如下:

(1) Makespan.反映任务完成时间,是衡量用户 QoS 保障的重要指标,取决于完成时间最大的计算节点.计算公式如下:

$$Makespan=\max\{CT\_node_i\} \tag{16}$$

其中,CT_node_i 表示节点 $node_i$ 上所有任务的完成时间.

(2) 系统利用率,即执行任务的节点数占所有节点数的比例.系统利用率越大,表示各节点的空闲资源得到越有效的利用,因而负载均衡性越好.计算公式如下:

$$utilization=exe\_node/all\_node \tag{17}$$

用户 QoS 方面主要指标有完成时间和费用,其中,费用已在用户目标函数 F_u 中有所体现(见第 2.2 节公式(7)及其说明),所以实验仅验证完成时间的表现;另外,对于不符合用户 QoS 和可用度极低的虚拟资源,采用了迁移、释放,因此对于减少能耗方面的效果比较明显;而负载均衡这一目标与系统利用率是一致的,因此实验仅验证在 3 个场景下的任务完成时间和系统利用率的表现.

4.2 基于熵优化和动态加权负载评估实验

为了验证本文提出的资源评估中两个改进点——熵优化原理和动态负载评估,设计了以下 3 个实验场景:(1) 仅采用熵优化原理的评估机制,用 EOWLEM1 表示;(2) 仅采用动态负载评估,用 EOWLEM2 表示;(3) 采用二者综合实验,用 EOWLEM3 表示.另外,实验又分节点数和任务数不同两种情况进行实验.

(1) 当节点数不同时,3 个场景、本文方法与其他 3 种方法对比的 Makespan 和系统利用率的表现如图 1、图 2 所示.

由图 1、图 2 可以看出,在 3 个场景中,仅采用熵优化原理的评估机制 EOWLEM1 在 Makespan 表现较优于仅采用动态负载评估 EOWLEM2,EOWLEM1 与 EOWLEM3 相差不大.其原因是:EOWLEM2 仅考虑了负载情况,没有顾及到用户 QoS,故在 Makespan 方面 EOWLEM1 与 EOWLEM3 表现较好;而 EOWLEM3 在与其他 3 种方法相比时优势较为明显,因为熵优化原理的计算复杂度低于 AEMJE,而且这种优势随着节点数的增多更加明显.

在系统利用率方面,EOWLEM2 优于 EOWLEM1.3 种场景中,EOWLEM3 表现最好,因为 EOWLEM3 采用熵优化保障了 QoS 及系统最大化,同时采用动态加权负载评估兼顾了负载均衡.而 EOWLEM3 与其他 3 种方法相

比,一样有明显优势.

无论 Makespan 还是系统,利用率表现最好的均是 EOWLEM3,皆因熵优化原理保障了用户 QoS(即对 Makespan 的贡献),动态加权负载评估提供了对系统利用率的表现.

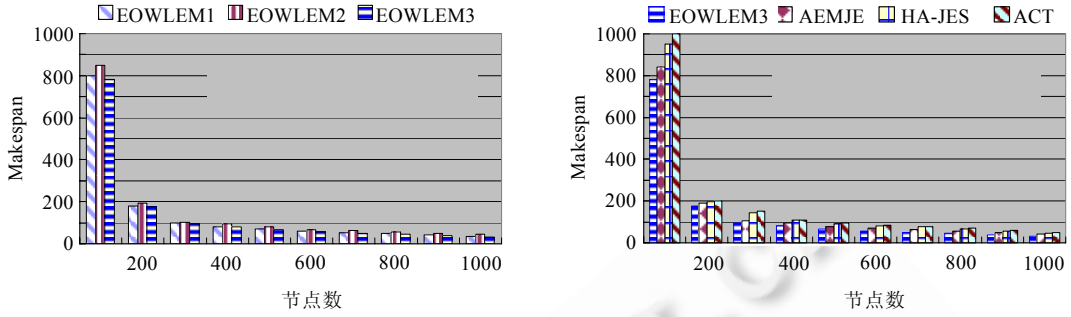


Fig.1 Makespan of different number of nodes

图 1 当节点数不同时,Makespan 的表现

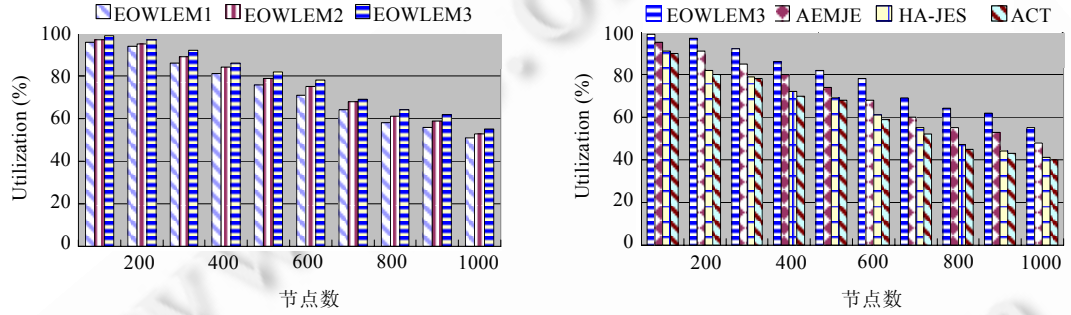


Fig.2 Utilization of different number of nodes

图 2 当节点数不同时,系统利用率的表现

(2) 当任务数不同时,3 个场景、本文方法与其他 3 种方法对比的 Makespan 和系统利用率的表现如图 3、图 4 所示.

当任务数不同时,各种方法的表现与节点数不同的情况类似,表现最好的仍然是 EOWLEM3,这也验证了熵优化模型对 Makespan、动态加权负载评估对系统利用率的贡献.

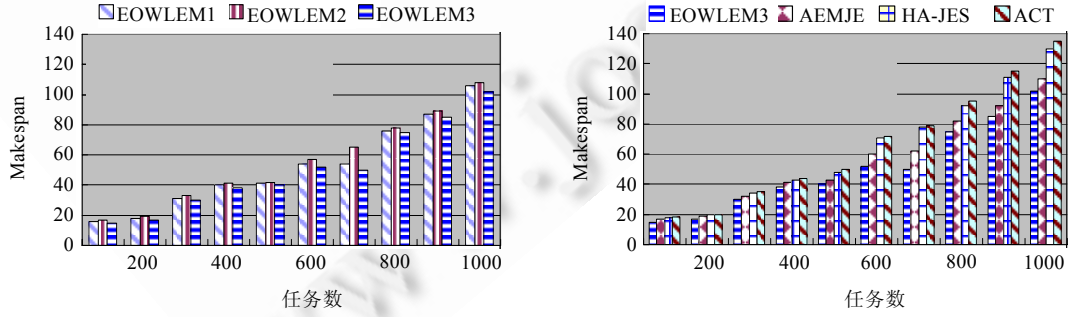


Fig.3 Makespan of different number of tasks

图 3 当任务数不同时,Makespan 的表现

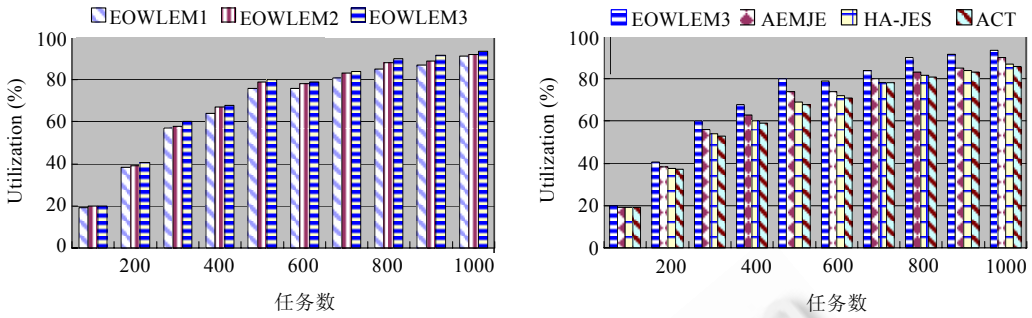


Fig.4 Utilization of different number of tasks

图 4 当任务数不同时,系统利用率的表现

5 结束语

为了准确地描述云计算中虚拟资源可用能力的动态特征,为调度或虚拟资源的迁移提供依据,实现云计算中的负载均衡、减少能耗、保障用户 QoS 和提高系统利用率等多目标优化,本文提出一种基于熵优化模型和动态加权负载评估的多目标优化评估机制.在熵优化模型中,通过最大熵和熵增最小原理,对满足用户 QoS 和系统最优目标函数的资源信息进行筛选,以保障用户 QoS 和系统最大化;采用动态加权负载评估算法对虚拟资源负载进行评估,对利用率不高及不满足熵优化的虚拟资源进行迁移、释放,实现负载均衡和减少能耗.为了验证该机制的性能,设计了评估实验,针对以上两个改进点分别实验,证明了熵优化模型在保障用户 QoS 和系统最大化方面起到一定的作用,而动态加权负载评估算法则在实现负载均衡和提高系统利用率方面颇为有效,尤其是在节点数和任务数比较多的情况下,本文评估机制相对于其他评估方法的优势更为明显,证明了该机制较适合云计算这样大规模的动态环境.

References:

[1] Iosup A, Jan M, Sonmez O, Epema DHJ. On the dynamic resource availability in grids. In: Proc. of the 8th IEEE/ACM Int'l Conf. on Grid Computing (Grid 2007). Texas: IEEE Computer Society, 2007. 26−33. [doi: 10.1109/GRID.2007.4354112]

[2] Khalili O, He J, Olsehanowsky C, Snavely A, Casanova H. Measuring the performance and reliability of production computational grids. In: Proc. of the 7th IEEE/ACM Int'l Conf. on Grid Computing (Grid 2006). Barcelona: IEEE Computer Society, 2006. 293−300. [doi: 10.1109/ICGRID.2006.311028]

[3] Xu M, Cui LZ, Wang HY, Bi YB. A multiple QoS constrained scheduling strategy of multiple workflows for cloud computing. In: Proc. of the 2009 IEEE Int'l Symp. on Parallel and Distributed Processing with Applications. 2009. 629−634. [doi: 10.1109/ISPA.2009.95]

[4] Chen K, Zheng WM. Cloud computing: System instances and current research. Ruan Jian Xue Bao/Journal of Software, 2009,20(5): 1337−1345 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/3493.html [doi: 10.3724/SP.J.1001.2013.03493]

[5] Tian WH, Zhao Y. Cloud Computing: Resource Scheduling Management. Beijing: National Defence Industry Publishing House, 2011 (in Chinese).

[6] Figueiredo R. Adaptive predictor integration for system performance prediction. In: Proc. of the IEEE Int'l Parallel and Distributed Processing Symp. IEEE Press, 2007. [doi: 10.1109/IPDPS.2007.370277]

[7] Diaz I, Fernandez G, Martinm M. Integrating the common information model with MDS4. In: Proc. of the 9th IEEE/ACM Int'l Conf. on Grid Computing. 2008. [doi: 10.1109/GRID.2008.4662812]

[8] Iosup A, Sonmez O, Epema D. The characteristics and performance of groups of jobs in grids. Lecture Notes in Computer Science, 2007,46(41):382−393. [doi: 10.1007/978-3-540-74466-5_42]

[9] Dinda PA, O'Hallaron DR. Host load prediction using linear models. Cluster Computing, 2000,3(4):265−280. [doi: 10.1023/A:1019048724544]

[10] Bucur AID, Epema DHJ. Scheduling policies for processor collocation in multicluster system. IEEE Trans. on Parallel and Distributed Systems, 2007,18(7):958−962. [doi: 10.1109/TPDS.2007.1036]

[11] Dai YS, Levitin G, Trivedi KS. Performance and reliability of tree-structured grid services considering data dependence and failure correlation. IEEE Trans. on Computers, 2007,56(7):925−936. [doi: 10.1109/TC.2007.1018]

[12] Fu S, Xu CZ. Exploring event correlation for failure prediction in coalitions of clusters. In: Proc. of the 2007 ACM/ IEEE Conf. on Super Computing (SC 2007). Nevada: IEEE Computer Society, 2007. 41−52. [doi: 10.1145/1362622.1362678]

[13] He SJ, Guo L, Guo YK, Ghanem M, Han R, Wu C. Elastic application container: A lightweight approach for cloud resource provisioning. In: Proc. of the 2012 IEEE 26th Int'l Conf. on Advanced Information Networking and Applications (AINA). IEEE, 2012. 15−22. [doi: 10.1109/AINA.2012.74]

[14] Kuehnhausen M, Frost VS, Minden GJ. Framework for assessing the trustworthiness of cloud resources. In: Proc. of the 2012 IEEE Int'l Multi-Disciplinary Conf. on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA). 2012. 142−145. [doi: 10.1109/CogSIMA.2012.6188367]

[15] Verma A, Cherkasova L, Campbell RH. Resource provisioning framework for MapReduce jobs with performance goals. Lecture Notes in Computer Science, 2011,70(9):165−186. [doi: 10.1007/978-3-642-25821-3_9]

[16] Katsaros G, Kousiouris G, Gogouvitis SV, Kyriazis D, Menychtas A, varvarigou T. A self-adaptive hierarchical monitoring mechanism for clouds. Journal of Systems and Software, 2012,85(5):1029−1041. [doi: 10.1016/j.jss.2011.11.1043]

[17] Gong ZH, Gu XH, Wilkes J. PRESS: Predictive elastic resource scaling for cloud systems. In: Proc. of the 2010 Int'l Conf. on Network and Service Management (CNSM). 2010. 9−16. [doi: 10.1109/CNSM.2010.5691343]

[18] Hu ZJ. The resource availability evaluation in service grid environment for QoS [Ph.D. Thesis]. Changsha: Central South University, 2010 (in Chinese with English abstract).

[19] Iosup A, Jan M, Sonmez OO, Epema DHJ. The characteristics and the performance of groups of jobs in grids. Lecture Notes on Computer Science, 2007,4641(8):382−393. [doi: 10.1007/978-3-540-74466-5_42]

[20] Ebrahimi N, Maasoumi E, Soofi ES. Ordering univariate distributions by entropy and variance. Journal of Econometrics, 1999, 90(2):317−336. [doi: 10.1016/S0304-4076(98)00046-3]

[21] Tao M, Dong SB, Zhang L. A multi-strategy collaborative prediction model for the runtime of online tasks in computing cluster/ grid. Cluster Computing, 2011,14(2):199−210. [doi: 10.1007/s10586-010-0145-4]

[22] Buyya R, Murshed M. Gridsim: A toolkit for the modeling and simulation of distributed resource management and scheduling for grid computing. Concurrency and Computation: Practice and Experience, 2002,14(12):1175−1220.

[23] Azougagh D, Yu JL, Kim JS, Maeng SR. Resource co-allocation: A complementary technique that enhances performance in grid computing environment. In: Proc. of the 1lth Int'l Conf. on Parallel and Distributed System (ICPADS 2005). Fukuoka: IEEE Computer Society, 2005. 36−42. [doi: 10.1109/ICPADS.2005.253]

附中文参考文献:

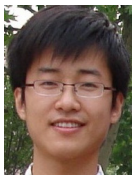
[4] 陈康,郑纬民.云计算:系统实例与研究现状.软件学报,2009,20(5):1337−1345. http://www.jos.org.cn/1000-9825/3493.html [doi: 10.3724/SP.J.1001.2013.03493]

[5] 田文洪,赵勇.云计算:资源调度管理.北京:国防工业出版社,2011.

[18] 胡周君.计算网格中面向 QoS 的资源可用性评估模型研究[博士学位论文].长沙:中南大学,2010.

左利云(1980－),女,河南周口人,副教授,主要研究领域为云计算,资源评估,调度.
E-mail: yuerly666@126.com

董守斌(1967－),女,博士,教授,博士生导师,主要研究领域为高性能计算.
E-mail: sbdong@scut.edu.cn



曹志波(1985－),男,博士生,主要研究领域为虚拟资源调度.
E-mail: caozhibo@126.com