数据中心中能量与 QoS 保障的虚拟机部署

王以伍 陈跃辉
(成都医学院网络中心 四川 成都 610083)

摘 要 针对已有虚拟机部署方法侧重于优化数据中心能耗,而忽略 QoS 保障的问题,提出一种基于能量与多维度 QoS 保障的虚拟机部署算法。建立虚拟机部署的 QoS 模型,设计一种通用 QoS 效用函数,实现不同形式 QoS 的标准量化。并在此基础上,将虚拟机部署问题形式化为满足全局 QoS 保障的同时能耗最小化问题。设计一种基于改进粒子群算法的虚拟机部署策略对优化问题进行求解。该策略通过相关参数和进化操作的重新定义,以及局部适应度优先的粒子位置更新机制,实现能耗与全局 QoS 保障的均衡优化。对算法进行了仿真实验分析,结果表明,该算法不仅在能耗与全局 QoS 保障性能上是优于同类算法的,并且在稳定性和可扩展性方面也具有较好的性能表现。

关键词 数据中心 虚拟机部署 QoS 保障 能耗优化

中图分类号 TP393 **文献标识码** A DOI:10.3969/j.issn.1000-386x.2019.05.010

VIRTUAL MACHINE DEPLOYMENT WITH ENERGY AND QOS GUARANTEE IN DATA CENTER

Wang Yiwu Cheng Yuehui
(*Center of Network, Chengdu Medical College, Chengdu 610083, Sichuan, China*)

Abstract To solve the problem that existing virtual machine deployment methods focus on optimizing the energy consumption of data centers while ignoring the QoS guarantee, we proposed a virtual machine deployment based on energy and multi-dimensional QoS guarantee. We established a QoS model for virtual machine deployment, designed a general QoS utility function, and realized the standard quantification of different forms of QoS. On this basis, we formalized the problem of virtual machine deployment as a problem of minimizing energy consumption while satisfying global QoS guarantee. Then we designed a virtual machine deployment strategy based on improved particle swarm optimization to solve this optimization problem. The strategy realized the balanced optimization of energy consumption and global QoS guarantees by redefining the relevant parameters and evolutionary operations, and updating the location of particulates with local fitness priority. And the algorithm was simulated and analyzed. The results show that the algorithm is superior to other similar algorithms not only in energy consumption and global QoS guarantee performance, but also in stability and scalability.

Keywords Data center Virtual machine deployment QoS guarantee Energy consumption optimization

0 引 言

随着云数据中心服务器数量与规模的迅猛增加,能耗已经成为数据中心面临的主要问题[1]。目前,数据中心的能耗已经占据全世界电力供应的 1.3%,至 2020 年将增长至 8%[2]。数据中心的碳排放为全球总量的 0.6%,等同于新西兰一个国家的碳排放量,至 2020 年将增长至 2.6%,超过德国的碳排放量。基于虚拟化的服务器合并技术是目前降低数据中心能耗的最重要手段,该技术使得同一服务器上可以运行多重应用,每种应用执行于各自的虚拟机上,并将虚拟机映

收稿日期:2018 - 10 - 08。成都市云平台建设基金项目(TY2014201)。**王以伍**,工程师,主研领域:云平台设计,信息化建设。**陈跃辉**,副教授。

射至服务器。此时,如何在满足服务质量 QoS 的同时设计出高能效的虚拟机部署算法对数据中心的能效将具有重要影响。

为了解决以上问题,本文在数据中心环境下提出了一种能量与 QoS 保障的虚拟机部署算法,算法的优势在于:1) 与传统的服务器资源同质不同,算法考虑了异构条件下的服务器资源,将从全局考虑 QoS 保障,将虚拟机部署问题形式化能耗与全局 QoS 保障间均衡的优化问题;2) 通过参数与进化操作的重定义,提出了一种改进的粒子群算法对虚拟机部署进行求解,并以局部适应度优先策略更新粒子位置,使算法可以更快收敛。

1 相关研究

已有研究中,MBFD 算法[3]是云数据中心中具有代表性的虚拟机部署能量优化算法。该算法首先按当前 CPU 占用对所有虚拟机进行降低排列,然后将虚拟机分配至带来功耗增加幅度最小的服务器上,实现能耗优化。此外,智能群体算法也广泛应用于虚拟机部署问题的求解,如遗传算法 GA[4]、模拟退火算法 SA[5]、粒子群算法 PSO[6]等。然而,以上已有工作均是同质的数据中心环境,其方法不适用于异构数据中心环境。此外,以最小化能耗为目标的单目标优化虚拟机部署也可以利用群体智能算法 ACO[7]、PSO[8]、SA[1]和 GA[9]等进行求解。但是,尽管此时算法得到的能耗有所降低,但在应用方面并未提供 QoS 保障,这使得应用负载的执行成功率不高。

考虑 QoS 保障的虚拟机部署问题的相关研究中,文献[10]提出了基于 GA 的多目标虚拟机部署算法,算法可以实现活动主机数量和通信 QoS 开销的最小,均衡数据中心中多维度资源的同步利用。文献[11]提出基于 GA 的虚拟机部署算法,可以最小化活动数量和最大化资源利用率。文献[12]则利用 PSO 实现了虚拟机部署问题中能耗最小和资源浪费最小。文献[13]利用一种文化基因算法实现虚拟机部署问题的多目标优化,包括最小化能耗、网络流量及最大化经济收益。问题在于,以上算法均只考虑了单个维度上的 QoS 保障,如通信开销、资源利用率、经济收益等,没有设计满足多维度 QoS 且从全局角度进行 QoS 保障的虚拟机部署算法。

比较已有研究,本文将设计能量与 QoS 保障的虚拟机部署算法,算法利用改进的粒子群优化可以在满足全局 QoS 保障的同时最小化数据中心的能耗。

2 能量与 QoS 模型

表 1 给出本文有关参数说明。

表 1 符号说明

符号	参数含义
ps_i	数据中心中第 i 个服务器主机,$i=1,2,\cdots$
vm_j	数据中心中第 i 个虚拟机,$j=1,2,\cdots$
w_k	第 k 个 QoS 属性的权重
S	服务
$Q_{j,k}^{\max}$	服务器 j 的第 k 个 QoS 属性的最大值
$Q_{j,k}^{\min}$	服务器 j 的第 k 个 QoS 属性的最小值
$q_j(S)$	服务的第 j 个属性值
r_i^{cpu}	虚拟机 i 的最大 CPU 请求
r_i^{mem}	虚拟机 i 的最大内存请求
c_j^{cpu}	服务器主机 j 的 CPU 能力
c_j^{mem}	服务器主机 j 的内存能力
$u_{ij}(t)$	运行于服务器主机 j 的虚拟机 i 的 CPU 利用率
X_i^t	位矢量,代表一个可行虚拟机部署方案
$u(t)$	随时间变化的 CPU 利用率
$P(u(t))$	时间 t 时服务器主机的功耗
$P_{\max}$	服务器主机满载时最大功耗
f	局部适应度函数
En	服务器主机的总体能耗

2.1 能耗模型

数据中心的能耗主要集中于服务器主机上,功耗主要由 CPU、内存、磁盘和网络接口组成,且 CPU 是其功耗的主要组成部分,也可认为,服务器的 CPU 利用率即为其资源利用率。CPU 利用率可根据执行负载的变化建立为时间的函数,故服务器的能耗也可基于 CPU 利用率进行建立。将服务器总体定义为:

$$En = \int_{t_1}^{t_2} P(u(t))\,\mathrm{d}t$$
$$P(u(t)) = c \times P_{\max} + (1-c) \times P_{\max} \times u(t) \tag{1}$$

式中:En 为时间段$[t_1,t_2]$间服务器的总能耗,$u(t)$为变化的 CPU 利用率,$P(u(t))$为时间 t 时服务器的功耗,$P_{\max}$为服务器满载时的最大功耗,c 为服务器为空闲状态时的能耗比例。

2.2 QoS 效用函数

数据密集型服务(简称服务)的 QoS 需求包括多种属性,如:响应时间、可靠性、吞吐量和可用性等。通常,这些属性可划分为两类:积极型 QoS 属性和消极型

QoS 属性。积极型 QoS 属性(如可靠性和可用性)表明其属性值越大,服务器运行相关服务的性能越好。相反,消极型 QoS 属性(如响应时间和延时)的属性值越低,性能越好。本文通过将积极型 QoS 属性转换为消极型 QoS 属性的方式(乘以 -1)而仅考虑消极型 QoS 属性。

考虑拥有 r 种 QoS 属性的服务 s 的 QoS 需求矢量 $\boldsymbol{qs}=\{q_1(s),q_2(s),\cdots,q_r(s)\}$,其中,$q_k(s)$值($1\leqslant k\leqslant r$)代表服务 s 的第 k 个属性值。所有 l 个服务的属性矢量可表示为 $\boldsymbol{QS}=\{q_1(S),q_2(S),\cdots,q_r(S)\}$,$S=\{s_1,s_2,\cdots,s_m\}$,其中,$q_k(S)$表示所有服务的第 k 个属性值的累加。表 2 给出了服务的 QoS 属性值的累加函数。

表 2　QoS 属性值的累加函数

QoS 属性	函数
响应时间	$q(S)=\max_{i=1}^{l}q(s_i)$
吞吐量	$q(S)=\sum_{i=1}^{l}q(s_i)$
可用性、可靠性	$q(S)=\min_{i=1}^{l}q(s_i)$

每种服务涉及多种 QoS 属性,会带来不同的量化程度,这不利于满足全局的 QoS 保障。因此,需要设计一种 QoS 效用函数,将 QoS 属性值 $q(s)$ 矢量映射为单一实数值。考虑 n 台服务器组成的数据中心,$PS=\{ps_1,ps_2,\cdots,ps_n\}$,可部署的 m 台虚拟机集合为 $VM=\{vm_1,vm_2\cdots,vm_m\}$。一种服务由部署于一台主机上一台虚拟机完成,同时需要满足其具体资源需求(即 CPU 和内存)和 QoS 约束。一种服务仅运行一台虚拟机上,且服务的 QoS 通常与虚拟机提供相关。因此,QoS 效用函数需要将所有属性值在多维度以统一计算的标准化方式将其映射至域[0,1]之间,从而量化全局的 QoS 保障性能。

定义 1　QoS 效用函数。假设有 r 个 QoS 属性,运行于第 j 台服务器 ps_j($1\leqslant j\leqslant n$)的虚拟机的第 i 个服务 $s_i\in S(1\leqslant i\leqslant l)$的 QoS 效用函数可表示为:

$$U(s_i)=\sum_{k=1}^{r}\frac{Q_{j,k}^{\max}-q_k(s_i)}{Q_{j,k}^{\max}-Q_{j,k}^{\min}}\cdot w_k \tag{2}$$

那么,所有服务 S 的 QoS 效用函数为:

$$U(S)=\sum_{k=1}^{r}\frac{Q_k^{\max}-q_k(s)}{Q_k^{\max}-Q_k^{\min}}\cdot w_k \tag{3}$$

同时:

$$\begin{cases}Q_k^{\max}=\sum_{k=1}^{r}Q_{j,k}^{\max}\left(Q_{j,k}^{\max}=\max\limits_{\forall s_i\in ps_j}q_k(s_i)\right)\\ Q_k^{\min}=\sum_{k=1}^{r}Q_{j,k}^{\min}\left(Q_{j,k}^{\min}=\min\limits_{\forall s_i\in ps_j}q_k(s_i)\right)\end{cases} \tag{4}$$

式中:$w_k\in\mathrm{R}^+$,$\sum_{k=1}^{r}w_k=1$,表示每种 QoS 属性的权重,用户可以根据其需求调整权重大小。$Q_{j,k}^{\max}$表示服务器 j 的第 k 个 QoS 属性的最大值,$Q_{j,k}^{\min}$表示服务器 j 的第 k 个 QoS 属性的最小值,$Q_k^{\max}$ 表示所有服务器上每种 $Q_{j,k}^{\max}$之和,$Q_k^{\min}$ 表示所有服务器上每种 $Q_{j,k}^{\min}$之和。

2.3　能量与 QoS 保障的虚拟机部署模型

数据中心的虚拟机部署问题的优化目标是在满足全局 QoS 保障的同时最小化能耗。通过重写式(1)和式(2),可以将能量与 QoS 保障的虚拟机部署问题形式化为多目标约束最优化问题,即带有全局 QoS 效用函数最大化问题(即全局 QoS 保障)的全局能耗最小化问题,定义为:

$$\min\sum_{j=1}^{n}\sum_{i=1}^{m}E_jx_{ij} \tag{5}$$

$$\max\sum_{k=1}^{r}\frac{Q_k^{\max}-\sum_{j=1}^{n}\sum_{i=1}^{m}x_{ij}\cdot q_k(s_i)}{Q_k^{\max}-Q_k^{\min}}\cdot w_k \tag{6}$$

约束条件包括 QoS 约束和资源能力约束,即:

$$\sum_{j=1}^{n}\sum_{i=1}^{m}x_{ij}\cdot q_k(s_i)\leqslant C_k\quad 1\leqslant k\leqslant r \tag{7}$$

$$\sum_{i=1}^{m}x_{ij}\cdot r_i^{\mathrm{cpu}}<c_j^{\mathrm{cpu}} \tag{8}$$

$$\sum_{i=1}^{m}x_{ij}\cdot r_i^{\mathrm{mem}}<c_j^{\mathrm{mem}} \tag{9}$$

$$\sum_{j=1}^{n}x_{ij}=1\quad i=1,2,\cdots,m \tag{10}$$

式中:n 表示数据中心中服务器的数量,m 表示部署虚拟机的数量,E_j 表示服务器 j 的能耗,C_k 表示 QoS 约束值,$C_k\geqslant q_k(S)$。

式(7)表明所有服务的 QoS 累加值不能超过 QoS 约束值,式(8)和式(9)表明虚拟机的资源请求不能超过服务器的资源能力,式(10)表明一个虚拟机只能部署至一个服务器,若虚拟机 i 运行于服务器 j,则 $x_{ij}=1$,否则 $x_{ij}=0$。同时,由于数据中心的资源异构性,c_j^{cpu} 与 c_k^{cpu} 不同,c_j^{mem} 与 c_k^{mem} 也不同。

式(5)-式(6)定义的虚拟机部署最优化问题为 NP 问题,该问题需要在满足所有约束(式(7)-式(10))的前提下实现最小化能耗和最大化全局 QoS 效用值,得到最优虚拟机部署方案。本文将提出一种基于改进粒子群算法的部署策略,在能量与全局 QoS 保障均衡下实现最优虚拟机部署。

3 算法设计

3.1 PSO 算法

粒子群算法 PSO 是一种基于群体智能的随机式搜索算法。作为一种进化计算技术,比较同类方法,PSO 执行速度更快,效率更高,已经广泛应用于人工智能神经网络训练、模糊系统控制等诸多领域。PSO 通过迭代式提供侯选解的方式不断改进解的质量,使其最终收敛于最优解。

PSO 中群体的每个成员即为一个粒子,每个粒子代表搜索问题的一个可行解。每个粒子拥有两个参数:速率和位置。粒子位置与适应度值相关,可用于评估解的质量。PSO 开始于随机产生的一个粒子群,并迭代寻找问题最优解。以 X_{lbest} 表示粒子找到历史最优解,X_{gbest} 表示整个种群找到的历史最优解,在每一次迭代中,粒子的速度和位置通过下式进行更新:

$$V_i^{t+1} = wV_i^t + c_1 r_1 \times (X_{\text{lbest},i}(t) - X_i^t) + c_2 r_2 \times (X_{\text{gbest},i}(t) - X_i^t) \tag{11}$$

$$X_i^{t+1} = X_i^t + V_i^{t+1} \tag{12}$$

式中:V_i^t 和 V_i^{t+1} 分别表示粒子在更新前后的速度,w 表示惯性权重因子,用来限定当前速度对粒子行为的影响,当 w 较大时,算法具有较强的全局搜索能力,当 w 较小时,算法具有较强的局部搜索能力;c_1 与 c_2 称为学习因子,分别用来限定个体经验和群体经验对粒子轨迹的影响,若 $c_1=0$,则粒子只向群体学习,这样容易陷入局部极值,若 $c_2=0$,则粒子只向自身学习,此时算法退化为一个多起点的随机搜索,通常,$c_1, c_2 \in [0,4]$;$r_i \in [0,1]$ 为随机数,$i=1,2$,X_i^t 和 X_i^{t+1} 分别表示粒子更新前后的位置。

为了求解能量与 QoS 保障的虚拟机部署问题,PSO 需作以下改进:1) 传统 PSO 仅适用于求解连续优化问题,不适用于求解离散优化问题,这表明 PSO 的相关参数和进化操作必须重新定义;2) 必须重新设计粒子更新策略和解码策略,以表达虚拟机与服务器间的映射关系。

3.2 改进 PSO

本文对传统 PSO 的改进主要针对两点:1) 重新定义了 PSO 的参数和进化操作,使其可以求解能量与 QoS 保障的虚拟机部署优化这类离散最优化问题;2) 采用了一种局适应度优先策略更新粒子位置信息。

3.2.1 相关定义

传统 PSO 仅适用于求解连续最优化问题,无法求解能量与 QoS 保障的虚拟机部署优化这类离散最优化问题。为此,需要重新定义 PSO 的相关参数和进化操作。

定义 2 粒子位置。粒子位置 $X_i^t = \{x_{i1}^t, x_{i2}^t, \cdots, x_{in}^t\}$ 重定义为 n 位矢量,代表一种可行的虚拟机部署方案,n 为粒子编码的长度,等于数据中心中服务器的数量。

定义 3 粒子速率。粒子速率 $V_i^t = \{v_{i1}^t, v_{i2}^t, \cdots, v_{in}^t\}$ 重定义为 n 位矢量,代表虚拟机部署的调整决策。V_i^t 指导粒子位置更新操作,驱动虚拟机部署向着最优解方向调整。矢量 V_i^t 中的每个位的值等于 0 或 1。若为 0,则表明对应服务器及其虚拟机已被重新评估适应度并作出调整,否则为 1。

定义 4 减法操作符。将减法操作符 Θ 重定义为用于计算两种虚拟机部署方案之差。对于 $X_i^t \Theta X_k^t$,若部署方案 X_i^t 中对应的位值等于 X_k^t 中对应的位值,则进行减法操作后其值为 1;否则等于 0。例如:$(1,1,1)\Theta(1,0,0)=(1,0,0)$。

定义 5 加法操作符。将加法操作符 $\oplus$ 重定义为粒子更新期间由于惯性速率、最优位置和全局最优位置改变导致粒子速率更新的操作。那么,$P_1V_1^t \oplus P_2V_2^t \oplus \cdots \oplus P_nV_n^t$ 代表在概率 $P_1, P_2, \cdots, P_n$ 下利用 V_1^t 和在概率 P_n 下利用 V_n^t 更新其粒子速率,并定义概率 P_i $\left(\sum_{i=1}^{n} P_i = 1\right)$ 为惯性权重因子。例如:$0.3(0,0,1) \oplus 0.7(0,1,0,1) = (0,\#,\#,1)$。第二个位值为 0 的概率为 0.3,为 1 的概率为 0.7。此时,"#"代表的值是不确定的,该位值也称为不确定位值,不确定位值会影响粒子速率更新。改进 PSO 中,定义在种惯性权重因子 P_{1i}、P_{2i}、P_{3i},随机产生于区域[0,1]间。

定义 6 乘法操作符。将乘法操作符 $\otimes$ 重定义为用于计算更新粒子位置。$X_i^t \otimes V_k^{t+1}$ 代表在任意时间 t 粒子位置矢量 X_i^t 以速率矢量 V_k^{t+1} 移动时的粒子更新操作。乘法操作符 $\otimes$ 的计算规则如下:1) 若速率矢量的位值为 1,则位置矢量中对应的位值不变;2) 若速率矢量的位值为 0,则位置矢量中对应的位值发生改变。例如:$(1,0,1,0) \otimes (1,1,0,0)$,$(1,0,1,0)$ 为位置矢量,$(1,1,0,0)$ 为速率矢量,速率矢量中第三和第四个位值均为 0,代表第三和第四个服务器中对应的虚拟机部署方案需要更新。

基于以上五种定义,可将式(11) - 式(12)转换为下式:

$$V_i^{t+1} = p_{1i}V_i^t \oplus p_{2i}(X_{\text{lbest},i}(t) \Theta p_{3i}(X_{\text{gbest}}(t) \Theta X_i^t)) \tag{13}$$

$$X_i^{t+1} = X_i^t \otimes V_i^{t+1} \tag{14}$$

3.2.2　局部适应度优先策略

传统的 PSO 采用随机选择策略,可能影响 PSO 的全局收敛,降低求解效率。为了增强解的质量,本文设计一种局部适应度优先策略更新粒子位置。

为了便于表述,将粒子的第一个维度上的每个位值称为局部位置。在一个最优时段$[t_1,t_2]$间运行于该服务器的所有虚拟机的 CPU 利用率被称为局部能量适应度,定义为:

$$f^{e}_{\mathrm{lbest},i}=\frac{1}{t_2-t_1}\int_{t_1}^{t_2}\Big(\sum_{j=1}^{m}u_{ij}(t)\Big)\mathrm{d}t \tag{15}$$

式中:$u_{ij}(t)$表示运行于服务器主机 j 的虚拟机 i 的 CPU 利用率,m 表示运行于服务器 j 上的虚拟机的总数量。

运行于服务器上的所有虚拟机的 QoS 属性之和称为局部 QoS 适应度,定义为:

$$f^{q}_{\mathrm{lbest},i}=\sum_{k=1}^{r}\frac{Q^{\max}_{j,k}-q_k(s_i)}{Q^{\max}_{j,k}-Q^{\min}_{j,k}} \tag{16}$$

式中:$q_k(s_i)$表示运行于服务器 j 上的虚拟机 i 的第 k 个 QoS 属性值,r 表示 QoS 属性总量,$Q^{\max}_{j,k}$表示服务器 j 的第 k 个 QoS 属性的最大值,$Q^{\min}_{j,k}$表示服务器 j 的第 k 个 QoS 属性的最小值。

基于式(15)和式(16),局部适度度定义为:

$$f_{\mathrm{lbest},i}=\frac{f^{e}_{\mathrm{lbest},i}+f^{q}_{\mathrm{lbest},i}}{r} \tag{17}$$

对于局部适应度优先策略,当 PSO 需要更新一个粒子的确定局部位置时,服务器上拥有最大适应度的虚拟机将具有更高的概率被选择去更新局部位置。局部适应度代表服务器的 CPU 利用率和 QoS 属性之和。

3.2.3　解码策略

为了改进解的效率,基于服务器与虚拟机间一对多的映射特点,设计了一种二维解码策略,如图 1 所示,其中,n 为服务器数量,m 为部署于同一服务器上的虚拟机的数量。

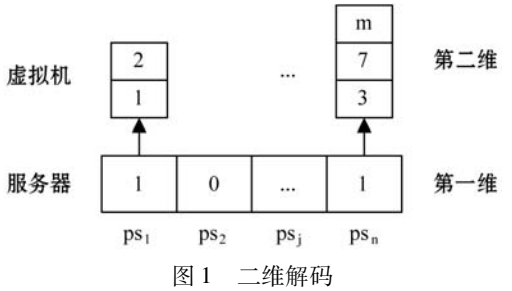

图 1　二维解码

由图可知,一个粒子的第一维为 n 位二进制矢量,矢量中的每一个位即对应于数据中心中的一个服务器。这里,"1"代表在当前虚拟机部署方案中对应服务器是活动的,"0"代表服务器为待机状态。一个粒子的第二维为部署于服务器上的虚拟机组成的集合。那么,每个虚拟机子集即与一个活动的服务器相关联。例如:粒子的第一维的第一个位值为 0,则表明数据中心的第一个服务器应被开启。第一、第二个虚拟机应部署于第一个服务器上。比较传统的一维粒子解码方法,二维解码策略不仅可以有效缩短粒子解码长度,降低搜索时间,而且能够反映虚拟机的静态优化部署特征。

4　仿真实验

4.1　实验配置

利用 CloudSim[14] 对算法进行仿真分析。模拟一个由 1 000 个异构服务器组成的数据中心环境,服务器具体由两类组成:一类是 HP Proliant G4,配置 CPU 为 3 720 MIPS,内存为 4 GB,峰值功耗为 117 W;另一类是 HP Proliant G5,配置 CPU 为 5 320 MIPS,内存为 4 GB,峰值功耗为 135 W。服务器具有不同的性能配置和能耗特征[3]。每个服务器运行带有四种 QoS 属性(响应时间、可用性、吞吐量和可靠性)的一个或多个数据密集型服务,服务负载产生现实的 2 500 个 Web 服务间。

为了更好地反映实际的虚拟机请求,以 Amazon EC2 的两类虚拟机实例作为虚拟机请求参数,包括 Micro 实例,配置 500 MIPS CPU 和 613 MB 内存,以及 Small 实例,配置 1 000 MIPS CPU 和 1 700 MB 内存。

选取改进最佳适应递减算法 MBFD[3]、最佳适应算法 BF[15]、能耗优化的粒子群算法 EOPSO[12] 作为性能比较的基准算法。前两种算法在装箱思想对虚拟机部署进行建模,区别在于对于服务器的资源占用排序分为递减和递增排列,两种算法在优化服务器使用数量、降低服务器闲置能耗上拥有较好效果。EOPSO 算法与本文同为粒子群优化,但其粒子进化操作及解码均与本文有所区别。其他参数配置如下:能量模型中参数 c 设置为 0.6,粒子群的初始种群设置为 20,粒子进化的最大迭代次数设置为 30,仿真实验运行 10 次,实验结果取其平均值。

4.2　结果分析

实验 1 观察数据中心所有活动服务器的总体能耗情况,结果如图 2 所示。可以看出,本文算法 EQPSO 比较其他算法节省了更多的能耗,且在虚拟机请求数量增加时,其能耗增加的敏感度也是最弱的。总体上,EQPSO 可以节省 30% 左右的能耗,原因在于 BF 和 MBFD 算法需要全局信息,即数据中心中异构服务器的

能耗信息,而仅仅考虑了多维度的资源约束,未考虑在虚拟机部署过程中不同服务器的能效的不同。PSOVMP 算法虽然也利用粒子群的求解思想,但其求解速度更慢,导致很多非最优解,能耗更大。本文算法引入了有效的粒子速率与位置更新机制,使其更易找到最优部署方案,也更易于实现算法收敛,改进了解的质量。最终,本文算法得到的活动服务器也较少,总体能耗更低。

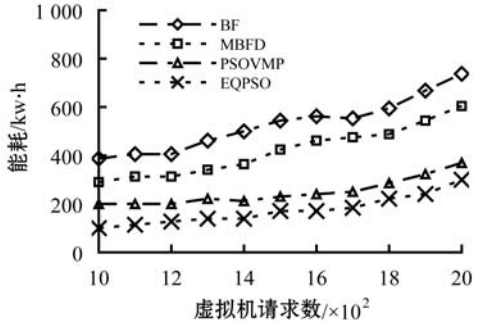

图 2 能耗

实验 2 观察全局 QoS 保障性能,结果如图 3 所示。由于四种 QoS 属性拥有不同的单位,需要根据 QoS 效用函数将 QoS 矢量值映射为单一实数值,即进行标准化后均映射至[0,1]之间,如此,全局 QoS 保障的取值范围为[0,4]。结果表明,本文算法 EQPSO 的全局 QoS 保障平均为 2.74,高于其他算法。原因在于其他算法侧重于局部 QoS 最优化,而局部 QoS 优化无法满足所有数据密集型服务的全局 QoS 保障。本文算法通过更低的能耗和更高的全局 QoS 保障获得了更好的性能。

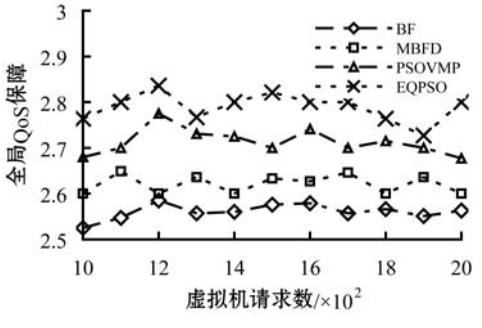

图 3 全局 QoS 保障

实验 3 观察服务器空闲的能耗比例变化对算法性能的影响,即式(1)中的 c 值,结果如图 4 所示。实验中设置 $c=0.1$,并以步长 0.1 递增至 0.9,同时将 QoS 约束的数量设置为 1,QoS 权重设置为 0.8。可以看出:1) 当 c 值递增时,能耗大幅增加,也反映出本文算法能耗越低时,c 值也越小,即空闲的服务器数量越多;2) 全局 QoS 保障并未随 c 值的变化发生剧烈波动。

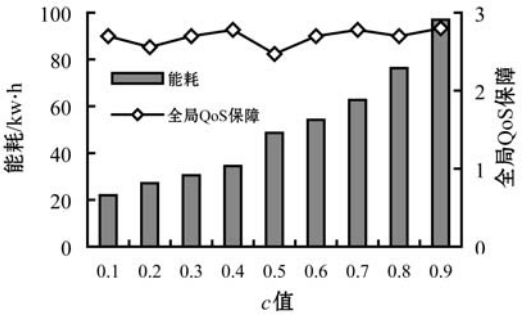

图 4 服务器空闲的能耗比例对性能的影响

实验 4 观察 QoS 约束数量对算法性能的影响,结果如图 5 所示。QoS 约束数量代表数据中心中用户对数据密集型服务的 QoS 需求。实验中设置 QoS 约束为 1,并以步长 1 递增至 4,c 设置为 0.6,w 设置为 0.8,其他属性权重随机产生于[0,0.2]之间。结果表明,本文算法得到的能耗与 QoS 保障并未随着 QoS 约束数量的变化发生过大波动,总体较为平衡,说明算法具有较好的鲁棒性,可适应于不同数量的 QoS 需求。

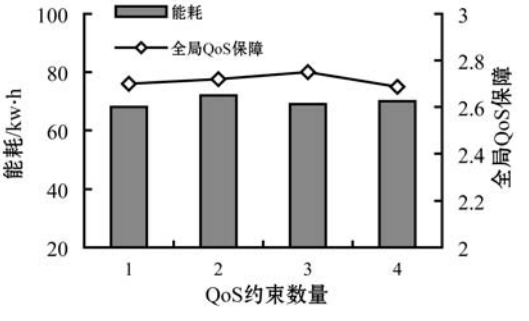

图 5 QoS 约束数量对性能的影响

实验 5 观察 QoS 权重 w 对算法性能的影响,结果如图 6 所示。实验中设置 $w=0.1$,并以步长 0.1 递增至 0.9,同时将 QoS 约束的数量设置为 1,c 设置为 0.6。结果表明:1) 全局 QoS 保障在 w 从 0.6 递增至 0.9 时也是递增的,即全局 QoS 保障性能越好,w 值越高;2) 算法能耗并未随着 w 值的变化发生剧烈波动,稳定性较好;3) 综合比较,在能耗与 QoS 均较优时,w 处于[0.7,0.9]间。

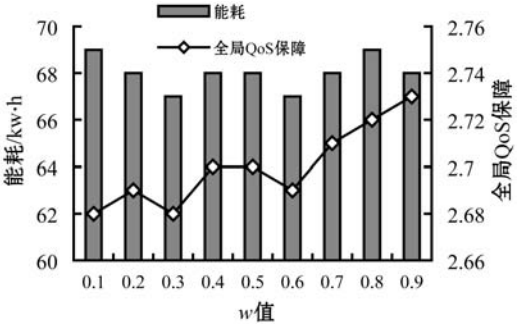

图 6 QoS 权重对性能的影响

实验 6 观察算法计算时间,结果如图 7 所示。实验中将虚拟机请求数量从 1 000 递增至 2 000,c 设置

为 0.6,w 设置为 0.8。结果表明,当虚拟机请求数量递增时,算法的计算时间增速较慢,计算时间与虚拟机请求数量之间几乎为线性关系,算法具体较好的可扩展性,可适应于不同规模的虚拟机部署请求。

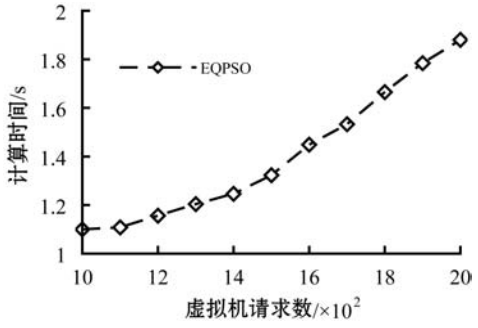

图 7　计算时间

5　结　语

提出了一种基于能量与多维度 QoS 保障的虚拟机部署算法。建立了虚拟机部署的 QoS 模型,设计了一种通用 QoS 效用函数,实现了不同形式 QoS 的标准量化,并在此基础上,将虚拟机部署问题形式化为满足全局 QoS 保障的同时实现能耗的最小化;设计了一种改进粒子群算法的虚拟机部署策略对优化问题进行求解,该策略通过相关参数和进化操作的重新定义,以及局部适应度优先的粒子位置更新机制,实现能耗与全局 QoS 保障的均衡优化。实验结果表明,算法不仅在能耗与全局 QoS 保障性能上是优于同类算法的,并且在稳定性和可扩展性方面也有较好的表现。

参　考　文　献

[1] Khalilzad N, Faragardi H R, Nolte T. Towards Energy-Aware Placement of Real-Time Virtual Machines in a Cloud Data Center[C]//International Conference on High PERFORMANCE Computing and Communications. IEEE Computer Society, 2015:1657 – 1662.

[2] Gao P X, Curtis A R, Wong B, et al. It's not easy being green[J]. ACM SIGCOMM Computer Communication Review, 2013, 42(4):211 – 222.

[3] Beloglazov A, Buyya R. Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in Cloud data centers[J]. Concurrency & Computation Practice & Experience, 2013, 24(13):1397 – 1420.

[4] Kaaouache M A, Bouamama S. Solving bin Packing Problem with a Hybrid Genetic Algorithm for VM Placement in Cloud[J]. Procedia Computer Science, 2015, 60(1):1061 – 1069.

[5] Wu Y, Tang M, Fraser W. A simulated annealing algorithm for energy efficient virtual machine placement[C]//IEEE International Conference on Systems, Man, and Cybernetics. IEEE, 2014:1245 – 1250.

[6] Wang S, Liu Z, Zheng Z, et al. Particle Swarm Optimization for Energy-Aware Virtual Machine Placement Optimization in Virtualized Data Centers[C]//International Conference on Parallel and Distributed Systems. IEEE, 2014:102 – 109.

[7] Liu X F, Zhan Z H, Du K J, et al. Energy aware virtual machine placement scheduling in cloud computing based on ant colony optimization approach[C]//Conference on Genetic and Evolutionary Computation. ACM, 2014:41 – 48.

[8] Xiong A P, Xu C X. Energy Efficient Multiresource Allocation of Virtual Machine Based on PSO in Cloud Data Center[J]. Mathematical Problems in Engineering, 2014, 20(5):1 – 8.

[9] Sharma K, Reddy M. Novel energy efficient virtual machine allocation at data center using Genetic algorithm[C]//International Conference on Signal Processing, Communication and Networking. IEEE, 2015:1 – 6.

[10] Liu C, Shen C, Li S, et al. A new evolutionary multi-objective algorithm to virtual machine placement in virtualized data center[C]//IEEE International Conference on Software Engineering and Service Science. IEEE, 2014:272 – 275.

[11] Joseph C T, Chandrasekaran K, Cyriac R. Improving the efficiency of genetic algorithm approach to virtual machine allocation[C]//International Conference on Computer and Communication Technology. IEEE, 2015:111 – 116.

[12] Ferdaus M H, Murshed M, Calheiros R N, et al. Virtual Machine Consolidation in Cloud Data Centers Using ACO Metaheuristic[C]//European Conference on Parallel Processing. Springer International Publishing, 2014:306 – 317.

[13] Pires F L, Benjam N. Multi-objective Virtual Machine Placement with Service Level Agreement: A Memetic Algorithm Approach[C]//IEEE/ACM, International Conference on Utility and Cloud Computing. IEEE Computer Society, 2013:203 – 210.

[14] Goyal T, Singh A, Agrawal A. Cloudsim: simulator for cloud computing infrastructure and modeling[J]. Procedia Engineering, 2013, 38(4):3566 – 3572.

[15] Babu K R R, Samuel P. Virtual Machine Placement for Improved Quality in IaaS Cloud[C]//Fourth International Conference on Advances in Computing and Communications. IEEE, 2014:190 – 194.