

功耗感知的自适应粒子群优化虚拟机动态映射

苏 宇^{1,2} 高 阳¹ 秦志光²

(南京大学计算机软件新技术国家重点实验室 南京 210093)¹

(电子科技大学信息与软件工程学院 成都 610054)²

摘 要 功耗管理是云计算数据中心的重要问题之一。由于服务器在不同睡眠状态时的功耗及唤醒延迟不同,将空闲服务器节电状态与输入作业负载建立映射,设计并实现了一种新的元启发式调度器,利用适应粒子群优化(SAPSO)检测和跟踪云计算资源池中不断变化的最优目标服务器,考虑了资源动态、工作服务器不同负载时的功耗、空闲服务器不同休眠状态转换时的功耗,使得 VM 映射中功耗增量最小。仿真实验表明了所提方法的有效性和较好的性能,经比较分析可知,该方法在保证满足 SLA 的情况下最大限度地减少了功耗且提高了 VM 映射效率。

关键词 虚拟机映射,功率感知,粒子群优化,云计算

中图分类号 TP301.6, TP18

文献标识码 A

DOI 10.11896/j.issn.1002-137X.2015.12.006

Power-aware Virtual Machine Dynamic Mapper Using Particle Swarm Optimization

SU Yu^{1,2} GAO Yang¹ QIN Zhi-guang²

(State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China)¹

(School of Information & Software Engineering, University of Electronic Science & Technology of China, Chengdu 610054, China)²

Abstract Power consumption management has become one of important issues in cloud computing. This paper designed and implemented a new meta-heuristic scheduler, in which the proposed self-adaptive particle swarm optimization (SAPSO) is used to detect and track the changing optimal target servers for virtual machine (VM) provisioning in the resource pool. The method considers resource dynamic and the power consumption of busy servers with different loads as well as idle servers in different sleep states, minimizing the incremental power in VM mapping due to workload mapping without degrading performance. Simulations show that the proposed power aware SAPSO is significantly effective to minimize the power consumption increment without compromising both the performances of the VM mapping and QoS with SLA to cloud users.

Keywords VM mapping, Power-aware, Particle swarm optimization (PSO), Cloud computing

1 引言

云计算作为一种抽象、虚拟化、动态可扩展的分布式计算模式,更是一种成本效益模型,它将计算、网络、存储、应用和服务等资源进行池化管理、弹性分配、按需供给,在满足云用户不断增长的应用需求的同时,需要对工作负载和资源配置进行快速智能优化和动态管理,从而实现能耗最小、虚拟机(VMs)性能最优的低成本、高效率的云服务运营模式。随着云计算的快速发展和应用普及,云计算数据中心的巨大功耗已成为当前亟待解决的重要问题,节能硬件、片上系统(SoC)架构和软件降耗技术给绿色计算带来新的机遇和挑战^[1,2]。数据中心节能降耗的技术和方法主要有虚拟化(Virtualization)^[2]、调节(Scaling)^[3]、调度(Scheduling)^[4]、整合(Consolidation)^[5]、优化(Optimization)^[6,7]等以及系统自动配置和电源接口(ACPI)管理、服务器动态整合、虚拟机在线迁移、存储

机制优化等。服务器功耗主要是通过动态构件失活(DCD)和动态电压频率调整(DVFS)的软件策略来动态改变 CPU 电压/频率进而实现优化的,动态的减速过程比全速空转能更有效地节能降耗。提高硬件资源利用率、微处理器的 DVFS 等,包括采用嵌入式智能电网技术以最优方式将电力传输到云计算设施,利用智能功耗节点管理器在高环境温度下进行功率和散热感知调度(PTAS),提高机架密度、主板电压调节、存储电源按需冗余等技术和方法^[7]。文献[8,9]结合 DVFS、DCD 的 IT 设备状态管理、电源管理策略,研究了 QoS 的异构集群中功耗感知的虚拟机迁移方法。文献[10]采用 CPU 频率调节技术,针对 CPU 的最佳频率来实现服务器最优功率分配且平均响应时间最小,以获得最佳的性能优化。DVFS 节能降耗的关键在于具体工作的执行时间的准确预测。

虚拟化由于其资源利用率高(服务器整合)、部署管理灵活(在线迁移)以及有隔离性、高可用性、可扩展性好等优点被

到稿日期:2015-02-02 返修日期:2015-07-23 本文受国家自然科学基金重点项目(61432008, 61133016),国家自然科学基金项目(61375121, 61075049),国家自然科学基金联合项目(U1435214, U1401257),国家科技重大专项(2011ZX03002-002-03),国家高技术研究发展 863 计划(2011AA010706)资助。

苏 宇(1993—),男,硕士生,主要研究方向为群智能大数据挖掘等;高 阳(1972—),男,博士,教授,博士生导师,主要研究方向为大数据机器学习、多 Agent 系统等;秦志光(1956—),男,博士,教授,博士生导师,主要研究方向为智能计算与信息安全、云计算与大数据处理等。

广泛应用。虚拟化使得不同用户的应用需求动态共享云计算物理资源,在保证服务质量(QoS)、满足服务协议(SLA)的同时,又提高了资源利用率且节省了功耗。VM中不同工作负载的CPU、内存、存储、网络密集型应用对系统功耗的响应特性不同,功耗的度量、建模、管理、优化等需要进一步研究^[1],峰值分配方法和平均值分配方法及过量分配策略会造成计算资源浪费和额外功耗。能效比(Performance/Power)是表示服务器能耗效率的重要指标^[12],动态整合虚拟机时优先整合能效比高的服务器,从而尽量减少虚拟机迁移次数和同时运行的服务器数量。

数据中心是支撑云服务环境的基础,使用功率感知的方法进行工作负载分配管理,从而在降低成本的同时实现更好地资源管理和更好的服务。当云用户希望明确SLA相关的计算资源以及服务器的复杂性和实时管理的增加时,性能和功耗感知的实时调度成为云计算数据中心具有挑战性的课题。成本开销、时间约束、负载类型、QoS偏好、计算资源效用、最小迁移代价的多目标VM放置优化等方法^[13-15]层出不穷。张等^[16]利用FFD尽可能地将虚拟机初始部署在低电压的物理机上,提出了一种功耗敏感的集群VM调度策略。近年来,研究者不断地引入求解约束VM映射问题的新型智能元启发式方法^[17-21],如FA用于VM负载平衡^[17],GA用于片上网络延迟约束下的针对功耗和温度的热感知^[18],考虑计算成本和数据传输成本,将PSO工作流调度方法应用于云资源配置^[19],引用ACO改善了异构环境下Hadoop资源感知调度器的作业性能^[20]。基于元启发式的方法,利用最小化新增VM的总功率增量的简单架构。在大规模分布式系统的能源任务调度中, Lee等^[14]考虑最小的完成时间和能源消耗,使用PSO搜索相对优势(Relative Superiority, RS)最大的处理器。现有处理器节能技术并没有考虑服务器的不同节电状态, Je-
yarani等^[21]设计了一种基于PSO的VM放置方法,在考虑服务器完成作业能力的同时,将空闲服务器节电状态与输入工作负载建立映射,由于不同状态下的功耗及唤醒延迟是不同的,使得功耗增量减少且响应更快。曾等^[22]通过虚拟机的高效迁移释放低载的物理主机并将其切入休眠状态,同时充分考虑被迁入目的主机的能耗影响,通过虚拟机之间的非合作博弈,以近似纳什均衡状态作为最终的迁移策略。

云计算环境中,数据中心节能管理需要考虑供给输入荷载的服务器特征来进行动态负荷预测^[22],本文设计并实现了一种新的元启发式调度器,利用粒子群优化(PSO)检测和跟踪云计算资源池中不断变化的最优目标服务器,在保证不损失性能和不延长响应时间的情况下,严格保证SLA的QoS的同时最大限度地减少功耗。

2 问题分析

要将 n 个VM实例放置于 m 个主机上,并达到性能最优、功耗最少。假定资源池中主机是动态异构的,每个计算资源中相应数量的处理机为虚拟机创建内存的大小、运行频率并且瞬时荷载都是使能DVFS的。计算资源根据负载情况可动态改变其状态,每个VM实例表示工作荷载所需的PE数和内存大小,而且负载相互独立、优先级相等。数据中心及其IT设施实际上是动态的,资源池的状态也在频繁变化,如虚拟化的热迁移、服务器会增加或减少可用的处理机数、内存大

小、存储容量和带宽等。

2.1 服务器的节电状态

状态管理是减少能耗的有效方法之一,状态管理是指根据负载的动态变化,在不同的情况下,令设备进入不同的节能状态,如休眠、关闭等。处于就绪状态的能耗有时甚至达到运行状态能耗的60%~80%,虚拟机迁移释放低载的物理主机,并将其切换到休眠状态,以达到降低能耗的目的。起初,一组计算资源根据要求处于浅休眠状态(SSS),其余的都处在深度休眠状态(DSS),资源池中主动PEs和空闲PEs可以根据瞬时负载灵活改变其运行频率,随着需求渐增,空闲PEs变为SSS状态;当需求渐少时,空闲PEs转换为DSS状态。不同休眠状态时服务器的节电和唤醒延时如表1所列。

表1 不同休眠状态时的节电和唤醒延时

休眠模式(S_i)	节能(%)	唤醒延迟
NAP(休眠)	20	5 μ s
SSS(浅休眠)	40	1ms
DSS(深休眠)	80	<100ms
IDLE(空闲)	100	<seconds

2.2 功耗模型

云计算数据中心的IT设施能耗主要来源于计算节点的CPU、内存、硬盘、网络等设备,即使在CPU利用率为10%时,功耗也超过峰值的50%,这导致非高峰时作业负载的功耗更多。CPU的当前功率与其效用呈线性关系,空闲状态的CPU的能耗相当于满载状态CPU的70%。肖等^[23]提出了一种基于状态管理的服务器功耗优化模型;Gandhi等^[23]研究了不同电源管理策略在多个服务器环境中的能耗性能和特点,假定服务器的负载到达率服从Poisson分布,而请求服务时间服从负指数分布,当不处理负载时服务器配置为不同休眠模式,但当负载到达时,系统切换到活动状态,这种电源模式切换产生相应的状态切换功耗和性能惩罚。若服务器的运行状态记为 S_{ON} ,各个空状态分别为 $S_0, \dots, S_N$,其中 S_N 为关闭状态,相应的功耗分别为 $P_{ON}, P_{S_0} = P_{IDLE}, P_1, \dots, P_{S_N} = P_{OFF}$ 。设 $P_{ON} > P_{IDLE} > \dots > P_{OFF} = 0$,从任何低电源状态 S_i 到高功率状态 S_j 需要切换时间 T_{S_i} ,其时长决定于目标状态,切换功耗上界为 P_{ON} ,而且 $T_{IDLE} < T_{S_1} < \dots < T_{S_{N-1}} < T_{OFF}$ 。当从高功率状态切换到低状态时,切换时延保持切换期间的功耗。如果服务器从低状态 S_i 以概率 p_i 切换到空闲状态,当一个负载到达服务器并不立即唤醒,而是以概率 q_{ij} 保持状态直到 j 作业进入等待队列,这段持续空闲时间称为恢复周期,它由3部分组成:

1)从休眠状态 S_i 直到 j 作业到达队列的时间,期望时延为服从Poisson分布的负载请求到达率 λ ,因此,这期间的功耗为 $(j/\lambda)P_{S_i}$;

2)唤醒期间 T_{S_i} 功耗: $T_{S_i}P_{ON}$;

3)当队列中所有 j 请求唤醒到 X 中,作业平均响应时间为 $1/(\mu - \lambda)$,对所有作业的期望时延为 $[(j+X)/(\mu - \lambda)]$,功耗为 $[(j+X)/(\mu - \lambda)]P_{ON}$,于是期望功耗是3段时间功耗之和:

$$E(P) = \sum_{i=0}^N p_i \sum_{j=1}^{\infty} \left[\frac{j}{\lambda} P_{S_i} + T_{S_i} P_{ON} + \frac{j + \lambda T_{S_i}}{\mu - \lambda} P_{ON} \right] + \sum_{i=0}^N p_i \sum_{j=1}^{\infty} q_{ij} \left[\frac{j}{\lambda} + T_{S_i} + \frac{j + \lambda T_{S_i}}{\mu - \lambda} \right] \quad (1)$$

其中, T_{S_i} 为Poisson分布的 X 的均值。

2.3 动态虚拟机映射问题

给定一个负载计算资源映射实例 M , 为了使得该映射到负载 (VM) 上的计算资源 (CR) 功率总增量最小, 同时满足最多工作负载的资源需求时性能最优。定义四元组 $C = \{CR, VM, P, M\}$, $CR(m, t) = \{CR_1, CR_2, \dots, CR_m\}$ 为一组 t 时刻可供的可用计算资源, 令 $VM(n, t) = \{VM_1, VM_2, \dots, VM_n\}$ 为时间窗 t_0 内的负载积累, 功耗模型 $P(m, t) = \{p_1, p_2, \dots, p_m\}$ 表示资源池中 (m) 计算资源的功耗。

轻负载的主动 PEs 优先分配, 调动的总功耗增量较小。当所有主动 PEs 满载时, 作业负载映射优先分配给 SSS 状态的 PEs, 因为相比分配给深睡眠服务器, 作业负载分配给浅睡眠服务器时总功耗增量更少, 要求至少一个映射满足 VM 请求的性能约束, 因此优化调度的目标是在所有的候选映射中实时找到一个映射实例 M 使得性能最优、功耗最小。

性能约束可用 VM 映射平均分配值 θ_r 和功耗来度量 $p(r, k)$, s 表示映射总数, $\theta_r \in (0, 1)$ 表示资源满足要求, s 个映射中具有最大值 θ_r 的 a 映射具有最好性能, 表示为 $M[m, n, a, t]$ 。

$$\theta_r = \left[\sum_{i=1}^n \sum_{j=1}^m \psi(i, j) / n \right] \quad (2)$$

$$i \in \{1, 2, \dots, n\}, j \in \{1, 2, \dots, m\}, r \in \{1, 2, \dots, s\}$$

$$\psi(i, j) = \begin{cases} 1, & \text{if } rrVM_i = arHost_j, i \in \{1, 2, \dots, n\} \\ 0, & \text{if } rrVM_i > arHost_j, j \in \{1, 2, \dots, m\} \end{cases} \quad (3)$$

其中, $\psi(i, j)$ 表示第 i 个作业负载在第 j 个主机上的一种分配, $rrVM_i$ 和 $arHost_j$ 分别表示第 i 个 VM 的最小资源要求和第 j 个主机的可用资源。

在 k 阶段的 a 成功映射表示为 $M[m, n, r, t + t_0(k)]$, 其相应的功耗为 $p(r, k)$, 其中映射数 $r \in \{1, 2, \dots, a\}$, k 表示一个连续时段, 那么在 $k-1$ 阶段分配的最好映射及其功耗分别为 $M[m, n, b, t + t_0(k-1)]$ 、 $p(b, k-1)$, 在所有计算资源 $CR(m, t)$ 中功耗增量 Δ_r , 当 Δ_r 最小时得到最优映射:

$$\begin{aligned} \Delta_r &= p(b, k-1) - p(r, k) \\ \delta_k &= \min\{\Delta_r\}, r \in \{1, 2, \dots, a\} \\ p(k) &= p(k-1) + \delta_k \end{aligned} \quad (4)$$

3 自适应粒子群优化

针对云计算环境的 VM 有效供给, 提出一种基于功耗感知的自适应粒子群优化算法。

3.1 粒子群优化

粒子群优化 (Particle Swarm Optimization, PSO) 算法是模拟鸟群寻找食物的集体行为而提出的一种基于群的全局优化技术。PSO 群中成员粒子 (并无质量和体积) 以一定的速度和加速度向更好的位置移动, 每个粒子表示问题空间的解。PSO 初始化为一群随机粒子, 每个优化问题的解看作搜索空间的一个粒子, 所有粒子都有一个由被优化的函数决定的适应值, 每个粒子还有一个速度用以决定它们飞行的方向和距离, 然后粒子追随当前的最优粒子在解空间中反复迭代搜索, 迭代中粒子通过跟踪两个“极值”即粒子本身所找到的最优解 $pbest$ 和群体找到的最优解 $gbest$ 来动态更新自己的速度和位置, 更新方程如下:

$$\begin{aligned} v(t+1) &= v(t) + c_1 r_1 (gbest - x(t)) + c_2 r_2 (pbest - x(t)) \\ x(t+1) &= x(t) + v(t+1) \end{aligned} \quad (5)$$

其中, v 为粒子速度, c_1, c_2 为加速因子, r_1, r_2 为 $(0, 1)$ 上均匀分布的随机数。每一维上的粒子速度被限制在 v_{max} ($v_{max} > 0$) 内, 如果某一维更新后的速度超过用户给定的 v_{max} , 那么速度就设为 v_{max} , 即 $v(t+1) > v_{max}$ 时, $v(t+1) = v_{max}$; 或 $v(t+1) \leq -v_{max}$ 时, $v(t+1) = -v_{max}$, 粒子从 $x(t)$ 到 $x(t+1)$ 的移动方向可能是 3 个方向的组合。PSO 由于其独特的搜索机制以及简单、高效、易于实现、潜在的并行性和分布式等特点, 已广泛应用于许多工程优化领域, 包括云服务中的虚拟机动态供给。

3.2 多集成粒子群优化 (MEPSO)

为了处理不断变化的动态环境, Du 等^[9]设计了一种多集成粒子群优化 (Multi-ensemble PSO, MEPSO), 粒子以概率 p 进行高斯 (Gaussian) 局部搜索, 提高了粒子在动态多峰环境下的寻优能力。

在动态搜索空间里, 粒子群初始分为两部分, 分别在搜索空间里使用不同策略。第一部分粒子尽可能快地探索当前环境中的全局最优, 为了提高第一部分群中粒子的局部搜索能力, 使用 Gaussian 搜索策略。迭代中粒子以概率 p_b 作 Gaussian 搜索 (见式 (7)), 而以 $(1 - p_b)$ 的概率作式 (6) 的搜索。第二部分粒子运动扩展了算法的搜索范围, 在第一部分周围跟踪变化的全局最优, 为此, 第二部分中粒子以概率 0.5 接近或远离选自群 1 中的个体最优。所谓 exploration 策略就是指差分变异并用式 (8) 更新速度, 以提高两群间的通信, 同时可避免在扩展粒子搜索区域时陷入早熟。

$$\begin{cases} v_{id}(t+1) = c_3 * rand() \\ x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \end{cases} \quad (6)$$

$$v_{id}(t+1) = \text{sgn}(r_1 - 0.5) \{wv_{id}(t) + c_1 r_2 [p_{id} - x_{id}(t)] + c_2 r_3 [p_{ad} - x_{id}(t)]\} \quad (7)$$

其中, $i=1, 2, \dots, m, d=1, 2, \dots, D, c_3$ 为正常数, 当 $k < 0, k=0, k > 0$ 时, $\text{sgn}(k)$ 分别取 $-1, 0, 1$ 。

3.3 动态自适应离散 PSO (DAPSO)

在动态环境中的全局最优和局部最优受随机定位的影响, 有两种情况: (1) $changedgbest$ 表示动态系统改变, 当前最优值也作相应改变; (2) 通过检测环境的变化重新评估全局最优值 $fixedgbest$ 。Carlisle 等^[3]提出一种带放哨粒子的自适应粒子群优化方法 (APSO), 用特定数目的放哨粒子 (Sentry Particle) 跟踪检测并自动响应粒子位置的局部状态变化。为了使 APSO 能适用于有噪声的分布式环境中, 后来 Carlisle 等^[3]又提出了一种能监测并自动响应环境变化的动态自适应 PSO (DAPSO)。DAPSO 可用于动态变化的噪声环境中找到移动的最优解, 它最有意义的特征是无集中控制的粒子记忆的恢复更新、及时重新评估适应值 $fitness$ 、更新个体最优 ($lbest$) 和全局最优 ($gbest$), 而且粒子的记忆更新没有任何集中控制, 也没有专门粒子监测环境的变化并没有额外的适应值评估计算使粒子适应变化了的环境, 如果当前适应值没有任何改进, 粒子个体最优 p_i 更新方程如下:

$$p_i(t+1) = \begin{cases} p_i(t) * E, & f(x_i(t+1)) = p_i(t) * E \\ f(x_i(t+1)), & f(x_i(t+1)) > p_i(t) * E \end{cases} \quad (8)$$

其中, 蒸发率常数 $E \in (0, 1)$ ^[5], 粒子适应值随着时间迭代以蒸发率 E 逐渐减少。如果粒子个体最优连续更新失败, 则其个体最优位置的适应值会逐渐减小, 最后个体最优和全局最

优适应值将低于当前粒子适应值。虽然所有的粒子具有相同的蒸发因子,但每个粒子的更新频率可能不一样,更新频率取决于粒子的个体最优适应值 $f(p)$ 和目前位置粒子适应值 $f(x)$ 。当 $f(p)$ 更小、 $f(x)$ 更大时,粒子将频繁更新其个体最优位置。但是在动态环境中当 $f(p)$ 更大、 $f(x)$ 更小时,则表示粒子远离最优解。通常地,新环境与之前环境越接近,之前的经历将更有利于搜索,适应值更新使得粒子能自适应环境的改变。

VM 供给的传统方法主要受性能目标驱使,并不适合于组合优化问题,应用自适应环境的感知技术如 DAPSO 来处理虚拟机配置管理,更适合于大规模高动态异构云计算环境,DAPSO 可以在合理的时间内得到近优解。

3.4 解的表示

n 个作业负载要分配到 m 个不同的计算资源上,PSO 粒子用 n 维向量表示,则每维上的值是一组可能的离散集合 $\{1, 2, \dots, m\}$,例如 3 个主机、10 个作业负载的情况,一组可能的映射实例解如表 2 所列,将最小位置值(SPV)^[10]的连续 PSO 算法应用于序列问题,粒子位置转换为排列序列, $s_i^k = (s_{i1}^k, s_{i2}^k, \dots, s_{im}^k)$,资源池中 m 个服务器,随机粒子数 $N \in (0, m)$ 。

表 2 PSO 粒子解的示例

维	x_{ij}^k	s_{ij}^k	r_{ij}^k	维	x_{ij}^k	s_{ij}^k	r_{ij}^k
0	1.13	3	3	5	3.51	8	0
1	3.91	9	1	6	2.74	5	1
2	-0.21	1	1	7	3.10	6	2
3	-0.39	0	0	8	0.99	2	2
4	3.25	7	3	9	1.62	4	0

3.5 功耗感知的启发式调度器

为了获取在一个时间窗 t_0 中 n 个待处理的工作负载表和可用计算资源数,该处理程序获取资源动态表如 PE 的计数、内存大小、瞬时功率消耗、主动和空闲服务器等所有可用的资源,然后调用 SAPSO,其实现步骤如图 1 所示。

算法 1 功耗感知的启发式 VM 调度器(metaScheduler)

获取资源池中服务器的动态功率状态

输入调度时间窗中作业负载表和作业负载数 n

输入资源池中当前功耗 P_0

Repeat

PSO 粒子向量的维= n

获取资源动态 CR;可用计算资源和当前节能状态

最佳映射实例=SAPSO(n, CRs, P_0)

分配作业

更新的资源池的资源动态

得到的作业负载表

Until(调度时间窗内无工作负载)

图 1 功耗感知的启发式 VM 调度器

3.6 自适应 PSO 算法

算法 PSO 初始化参数如表 3 所列,初始位置和初始速度是随机设置的。评估粒子适应值表示执行映射满足 VM 请求的最大数、计算平均分配值 θ_i 。首次迭代时,个体最优 p_i 设为当前值,在迭代中,检测和跟踪资源池的变化,蒸发率常数 E 用于删除过时的个体最优,因此在自适应供给中,个体适应值以蒸发率 E 随时间减少。如果粒子个体最优连续更新失败,则其个体最优位置的适应值会逐渐减小,最后个人最优将由对应变化环境中的当前值取代。因此该算法能感知新

开或关闭主机并更新资源供给信息,总是利用更好的资源,而且分配资源给不可用的主机的可能最小。算法中粒子保存个体最优,直到其最优适应值过时。适应值的更新评估帮助捕捉资源状况的变化如内存的增加或减少、可用核数等。最后,群中粒子对应的全局最优就是计算资源上的最优 VM 映射,VMs 被放置在这样的映射上,具体步骤如图 2 所示。

表 3 SAPSO 参数设置

参数	值
粒子数	20
维	10
c_1	2
c_2	1.92
w	[0.9, 0.4]
r_1, r_2	(0,1)
i	(0,50)
j	(-50,50)

算法 2 SAPSO (n, CRs, P_0)

初始化参数

设置资源动态区间

设置位置矢量 $x(i, k)$

设定速度向量 $v(i, k)$

计算序列向量 si 、操作矢量 r_k

用式(6)、式(7)评估粒子适应值

设置粒子个体最优 $lbest_j$,并作为当前位置

搜索群中全局最优 $gbest_j$;

设置 $k=2$;

Repeat

获取资源动态

for 群中每个粒子 i //并行程序执行

ParBegin

find_PersonalBest($v(i, k), x(i, k), i$)

ParEnd

搜索最大适应值 $Max\{\theta\}$

if ($Max\{\theta\}$ 多于 1 个)

用式(8)搜索功耗最小 Δ

更新 $globalBest$ 全局适应值= f_i

更新全局粒子位置 $globalBest = x(i, k)$

更新惯性权重 $w = w * 0.01$

$k = k + 1$;

Until($k > k_{max}$)

图 2 提出的自适应粒子群优化算法(SAPSO)

算法 3 Procedure find_PersonalBest(v, x, i)

Begin

更新粒子速度 $v(i, k)$

更新粒子位置 $x(i, k)$

应用 SPV 规则搜索解序列 $s(k)$;

if ($m < n$) $r(i, k) = s(i, k) \bmod m$ else $r(i, k) = s(i, k)$

用式(6)、式(7)计算粒子适应值

if ($f(r(i, k)) > f(p_i) * E$)

更新个体最优适应值 $F_i = f(r(i, k))$

更新个体最优位置 $p_i = x(i, k)$

else

$F_i = F_i * E$

End

图 3 SAPSO 中个体最优过程

实验使用事件驱动的模拟器 CloudSim 工具包,模拟动态环境下,利用相关抽样方差衰减技术。随机数发生器的周期长度为 109,资源池中有 100 个服务器,并设置不同大小的内存和不同数目的处理器(核数)。假定有 10 个批次的虚拟机要求,每个包含不同用户的 20 个请求和不同的资源需求。实验中调用本文提出的模块作为优化器,周期性地获取数据中心的资源状态。

4.1 检测和跟踪的最优目标服务器

实验目的是在追踪和发现放置 VM 的最优目标主机时验证 SAPSO 算法的有效性。在这种情况下,初始化负载运行时间,VMs 定期关闭选择,重新请求分配 PEs 和内存,这使得轻负载的节点(最佳)在随后的供给时间间隔内变为可用。使用 CloudSim 中存在变化的最优,在获取新一批的 VM 请求后,用 MEPSO 和 SAPSO 感知和跟踪不断变化的最优放置(VM)主机,从而减少功耗,而 StdPSO 方法实验却未成功,实验结果如图 4 所示。

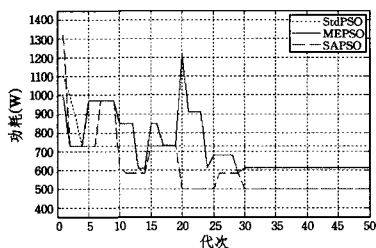


图4 3种算法性能比较

4.2 VM 供给失败率分析

在资源配置时距内,通过调度几个主机故障事件和主机关闭事件,模拟动态资源撤出。这些事件引起了一些虚拟机配置失败,因为部分 VM 请求映射的主机不可用。用 StdPSO 作为供给器时导致虚拟机分配失败次数更多,而用 MEPSO 或 SAPSO 在动态资源池中能更好地找到合适的主机,如图 5 所示。MEPSO 与 SAPSO 有较高的供给成功率,由于其在供给间隔内能够相当有效地检测到撤出的主机,并且找到 VM 资源请求的可用主机。

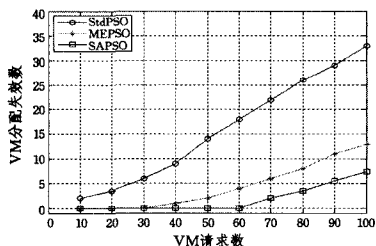


图5 VM需求增加时对算法性能的影响

结束语 随着云计算资源池规模的增大,用户对资源的突发需求越来越大,启发式的元调度器可以在短时间内提供合理的解决方案。用自适应 PSO 来感知并有效地利用资源状态的变化、资源动态和不同负载工作服务器的功耗,以及在不同睡眠状态的空闲服务器等情况下做出调度决策,从而在不降低 VM 映射性能的情况下最大限度地减少功耗增量。实验表明提出的基于功耗感知的 SAPSO 算法的有效性。要实现低成本、高效、安全、易用的云计算系统,依然面临诸多挑战,其中高能耗已成为云计算不可忽视的问题,高效、实时、动态预测的混合群智能资源调度器还需要进一步研究。

- [1] 叶可江,吴朝晖,姜晓红,等. 虚拟化云计算平台的能耗管理[J]. 计算机学报,2012,35(6):1262-1285
Ye K J, Wu Z H, Jiang X H, et al. Power management of virtualized cloud computing platform[J]. Chinese Journal of Computers, 2012, 35(6): 1262-1285
- [2] Zhuravlev S, Saez J C, Blagodurov S, et al. Survey of Energy-Cognizant Scheduling Techniques[J]. IEEE Trans Parall Distr Syst, 2013, 24(7): 1447-1464
- [3] Dargie W, Schill A. Analysis of the power and hardware resource consumption of servers under different load balancing policies[C]//Proc IEEE 5th Int Conf CLOUD, 2012:772-778
- [4] Merkel A, Stoess J, Bellosa F. Resource-Conscious Scheduling for Energy Efficiency on Multicore Processors[C]//Proc 5th Euro Sys, 2010:153-166
- [5] Srikantiah S, Kansal A, Zhao F. Energy Aware Consolidation for Cloud Computing[C]//Proc USENIX workshop Power Aware Comput Syst, 2008
- [6] Belogazov 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 and Computation: Practice and Experience, 2012, 24(13): 1397-1420
- [7] 向洁,丁恩杰. 基于虚拟机调度的数据中心节能优化[J]. 计算机应用, 2013, 33(12): 331-335
Xiang J, Ding E J. Energy saving optimization in datacenter based on virtual machine scheduling [J]. Computer Applications, 2013, 33(12): 333-335
- [8] 邓维,廖小飞,金海. 基于虚拟机的数据中心能耗管理机制[J]. 中兴通信技术, 2012, 18(4): 15-18
Deng W, Liao X F, Jin H. Energy Management Mechanisms in Virtualized Data Centers[J]. ZTE Technology Journal, 2012, 18(4): 15-18
- [9] Verma A, Ahuja P, Neogi A. PMapper: power and migration cost aware application placement in virtualized systems[C]//Proc Int Conf ACM Middleware, Springer-Verlag, New York, USA, 2008: 243-264
- [10] 陈昊罡,汪小林,王振林,等. DMM: 虚拟机的动态内存映射模型[J]. 中国科学:信息科学, 2010, 40(12): 1543-1558
Cheng H W, Wang X L, Wang Z L, et al. DMM: Dynamic Map Models of Virtual Machine[J]. Chinese Science: Information Science, 2010, 40(12): 1543-1558
- [11] 王桂彬,杨学军,徐新海,等. 异构系统功耗感知的并行循环调度方法[J]. 软件学报, 2011, 22(9): 2222-2234
Wang G B, Yang X J, Xu X H, et al. Power-Aware Parallel Loop Scheduling Method for Heterogeneous System[J]. Journal of Software, 2011, 22(9): 2222-2234
- [12] Carlisle A, Dozler G. Tracking changing extrema with adaptive particle swarm optimizer [C]//Proc 5th Biannual World Automation Congress, Orlando, Florida, USA, 2002: 265-270
- [13] Prado R P, Hoffmann F, García-Galán S, et al. On providing quality of service in grid computing through multi-objective swarm-based knowledge acquisition in fuzzy schedulers[J]. International Journal of Approximate Reasoning, 2012, 53(2): 228-247
- [14] Sheikh H F, Tan H X, Ahmad I, et al. Energy and performance-aware scheduling of tasks on parallel and distributed systems [J]. ACM Journal on Emerging Technologies in Computing Sys-

- [15] 张鲁飞, 陈左宁. 虚拟集群上面向功耗的形式化的 VM 调度策略[J]. 计算机科学, 2014, 41(8): 38-41, 46
Zhang L F, Cheng Z L. Power-efficient Formal Scheduling Policy of VMs in Virtualized Clusters[J]. Computer Science, 2014, 41(8): 38-41, 46
- [16] Koulinas G, Kotsikas L, Anagnostopoulos K. A particle swarm optimization based hyper-heuristic algorithm for the classic resource constrained project scheduling problem[J]. Information Sciences, 2014, 277(4): 680-693
- [17] Paulin Florence A, Shanthi V. A load balancing model using firefly algorithm in cloud computing[J]. Journal of Computer Science, 2014, 10(7): 1156-1165
- [18] Du WL, Li B. Multi-strategy ensemble particle swarm optimization for dynamic optimization [J]. International Journal of Information Sciences, 2008, 178(11): 3096-3109
- [19] 李进超, 陈静怡, 吴杰, 等. 基于改进分组遗传算法的虚拟机放置研究[J]. 计算机工程与设计, 2012, 33(5): 2053-2057
Li J C, Cheng J Y, Wu J, et al. Virtual machine placement research based on improved grouping genetic algorithm[J]. Computer Engineering and Design, 2012, 33(5): 2053-2057
- [20] Jeyarani R, Nagaveni N, Vasanth R. Design and implementation of adaptive power-aware virtual machine provisioner (APA-VMP) using swarm intelligence[J]. Future Generation Computer Systems, 2012, 28(4): 811-821
- [21] Xu G H, Ding Y, Zhao J, et al. A Novel Artificial Bee Colony Approach of Live Virtual Machine Migration Policy Using Bayes Theorem [J/OL]. <http://www.hindawi.com/journals/twsj/2013/369209>
- [22] Gao Y Q, Guan H B, Qi Z W, et al. A multi-objective ant colony system algorithm for virtual machine placement in cloud computing[J]. Journal of Computer and System Sciences, 2013, 79(8): 1230-1242
- [23] 张海燕, 吴凡, 王建新. 基于蚁群算法的 Hadoop 资源感知调度器研究[J]. 计算机工程与应用, 2014, 50(15): 65-71
Zhang H Y, Wu F, Wang J X. Research of Hadoop resource-aware scheduler based on ant colony algorithm[J]. Computer Engineering and Applications, 2014, 50(15): 65-71
- [24] 曾凯, 余堃, 敬思远. 云环境下基于功耗感知的虚拟机博弈迁移算法[J]. 计算机应用研究, 2013, 30(6): 1668-1671
Zeng K, She K, Jing X Y. Game theoretic migration algorithm based on power-aware for virtual machines in cloud[J]. Application Research of Computers, 2013, 30(6): 1668-1671
- [25] 宋杰, 李甜甜, 闫振兴, 等. 一种云计算环境下的能效模型和度量方法[J]. 软件学报, 2012, 23(2): 200-214
Song J, Li T T, Yan Z X, et al. Energy-Efficiency Model and Measuring Approach for Cloud Computing[J]. Journal of Software, 2012, 23(2): 200-214
- [26] Dong J K, Wang H B, Li Y Y, et al. Virtual Machine scheduling for improving energy efficiency in IaaS cloud[J]. China Communication, 2014, 11(1): 1-12
- [27] 赵淦森, 虞海, 季统凯, 等. 云计算平台的自适应资源供给[J]. 电信科学, 2012, 28(1): 31-37
Zhao G S, Yu H, Ji T K, et al. Adaptive Resource Provisioning for Cloud Computing[J]. Telecommunications Science, 2012, 28(1): 31-37
- [28] 肖志娇, 明仲, 蔡树彬. 基于状态管理的服务器节能策略研究[J]. 计算机科学, 2013, 40(4): 22-26
Xiao Z J, Ming Z, Cai S B. Study on Energy Optimization of Servers Based on States Management[J]. Computer Science, 2013, 40(4): 22-26
- [29] Möbius C, Dargie W, Schill A. Power Consumption Estimation Models for Processors, Virtual Machines, and Servers[J]. IEEE Trans Parall Distr Syst, 2014, 25(6): 1600-1614
- [30] Ansari KH, Chitra P, Sonaiya Karthick P. Power-Aware Scheduling of Fixed Priority Tasks in Soft Real-Time Multicore Systems[C]//IEEE Int Conf Emerging Trends in Computing, Communication and Nanotechnology (ICECCN). 2013: 496-502

(上接第 12 页)

- [30] Rietveld S, Oud M, Dooijes E. Classification of asthmatic breath sounds: preliminary results of the classifying capacity of human examiners versus artificial neural networks[J]. Computers and Biomedical Research, 1999, 32(5): 440-448
- [31] Waitman L R, et al. Representation and classification of breath sounds recorded in an intensive care setting using neural networks[J]. Journal of Clinical Monitoring and Computing, 2000, 16(2): 95-105
- [32] Alsmadi S S, Kahya Y P. Online classification of lung sounds using DSP[C]//Proceedings of the Second Joint Engineering in Medicine and Biology, 2002. 24th Annual Conference and the Annual Fall Meeting of the Biomedical Engineering Society EMBS/BMES Conference, 2002. IEEE, 2002, 2: 1771-1772
- [33] Güler İ, Polat H, Ergün U. Combining neural network and genetic algorithm for prediction of lung sounds[J]. Journal of Medical Systems, 2005, 29(3): 217-231
- [34] Matsunaga S, et al. Classification between normal and abnormal respiratory sounds based on maximum likelihood approach[C]//IEEE International Conference on Acoustics, Speech and Signal Processing, 2009 (ICASSP 2009). IEEE, 2009: 517-520
- [35] Yamashita M, Matsunaga S, Miyahara S. Discrimination between healthy subjects and patients with pulmonary emphysema by detection of abnormal respiration[C]//2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2011: 693-696
- [36] Abbas A, Fahim A. An automated computerized auscultation and diagnostic system for pulmonary diseases[J]. Journal of medical systems, 2010, 34(6): 1149-1155
- [37] İçer S, Gengeç Ş. Classification and analysis of non-stationary characteristics of crackle and rhonchus lung adventitious sounds [J]. Digital Signal Processing, 2014, 28: 18-27
- [38] Bahoura M. Pattern recognition methods applied to respiratory sounds classification into normal and wheeze classes[J]. Computers in Biology and Medicine, 2009, 39(9): 824-843
- [39] Liu Yi, et al. The Feature Extraction and Classification of Lung Sounds Based on Wavelet Packet Multiscale Analysis[J]. Chinese Journal of Computers, 2006, 29(5): 769-777
- [40] Lu Qiang, et al. Study for lung sound acquisition module based on ARM and Linux[J]. Chinese Journal of Medical Instrumentation, 2011, 35(4): 263-265
- [41] Sosa Ramírez G D, Ramírez G D S, Clavijo F V. Design and development of mobile application to characterize and classify lung sound based on Frequential Analysis and Wavelet Transform[J/OL]. http://www.researchgate.net/publication/272745174_application_movil