

面向云环境中任务负载的粒子群优化调度策略

胡志刚¹, 常健¹, 周舟^{2†}

(1. 中南大学 计算机学院, 湖南 长沙 410075;

2. 长沙学院 计算机工程与应用数学学院, 湖南 长沙 410022)

摘要:随着云环境中任务规模的不断扩大, 云计算中心高能耗问题变得日益突出. 如何解决云环境中任务分配问题从而有效降低能耗, 本文提出了一种改进的粒子群优化算法 (Modified Particle Swarm Optimization, M-PSO). 首先构建出一个云计算能耗模型, 同时考虑处理器的执行能耗和任务传输能耗. 基于该模型, 对任务分配问题进行定义描述, 并采用粒子群优化算法对问题进行求解. 此外, 构建动态调整的惯性权重系数函数以克服标准 PSO 算法的局部最优和收敛速度慢的问题, 有效提高系统性能. 最后通过仿真实验对该算法模型的性能进行了评估, 结果表明 M-PSO 算法与其他算法相比能有效地降低系统总能耗.

关键词:云计算; 任务调度; 惯性权重; 粒子群优化

中图分类号:TP338.8

文献标志码:A

PSO Scheduling Strategy for Task Load in Cloud Computing

HU Zhigang¹, CHANG Jian¹, ZHOU Zhou^{2†}

(1. School of Computer Science and Engineering, Central South University, Changsha 410075, China;

2. Department of Mathematics and Computer Science, Changsha University, Changsha 410022, China)

Abstract: As the scale of tasks in the cloud environment continues to expand, the problem of high energy consumption in cloud computing centers has become increasingly prominent. In order to solve the problem of task assignment in a cloud environment and to effectively reduce energy consumption, a Modified Particle Swarm Optimization algorithm (M-PSO) was proposed. First, a cloud computing energy consumption model, which takes into account the processor's execution energy consumption and task transmission energy consumption, was introduced. Based on the model, the task assignment problem was defined and described, and the particle swarm optimization algorithm was used to solve this problem. In addition, a dynamically adjusted inertia weight coefficient function was constructed to overcome the local optimization and slow convergence problem of the standard PSO algorithm, and the strategy can effectively improve the system performance. Finally, the performance of the introduced algorithm model was evaluated by simulation experiments. The results show that the M-PSO algorithm can effectively reduce the total energy consumption of the system compared with other algorithms.

Key words: cloud computing; task scheduling; inertia weight; Particle Swarm Optimization (PSO)

* 收稿日期:2018-07-11

基金项目: 国家自然科学基金资助项目 (61572525), National Natural Science Foundation of China (61572525); 长沙市科技计划资助项目 (k1705036), Science and Technology Plan Project of Changsha City (k1705036); 湖南省自然科学基金青年基金项目 (2019JJ50689), Natural Science Foundation of Hunan Province (2019JJ50689); 中国博士后科学基金项目 (2018M642974), China Postdoctoral Science Foundation (2018M642974); 湖南省教育厅优秀青年基金项目 (18B412), Scientific Research Project of Education Department of Hunan Province (18B412)

作者简介: 胡志刚 (1963—), 男, 山西孝义人, 中南大学教授, 博士生导师

† 通讯联系人, E-mail: zhouzhou03201@126.com

云计算提供了方便和按需的网络访问共享和可扩展的虚拟资源池,如服务器、存储和应用服务^[1]。但是在配电和散热方面,云计算数据中心会产生巨大的能源成本^[2]。对于像 Google 这样的大公司来说,能源成本降低三分之一可以节约成本超过一百万美元。与此同时,政府机构继续推行节能计算^[3]。因此,降低能源消耗已经成为当今数据中心运营商关注的主要问题。

针对云计算数据中心的能耗问题,文献[4-5]分别提出了一种精确度高的能耗模型来预测云计算数据中心单台服务器的能耗状况,从而实现对整个数据中心的能耗估测。如何在能耗与性能两者间取得平衡,一种典型的思路就是不定期检测系统中的空闲设备,然后将其关闭实现节约能耗的目标^[6]。但这种策略的主要问题在于:硬件设备从关闭状态重新启动达到运行状态,这一过程期间会导致系统性能的下降。而云计算的主要支持技术是虚拟化,以经济高效的方式将单个物理机器分割为多个虚拟机(Virtual Machine, VM),这种技术可以有效地将物理设备逻辑上分成若干子模块,从而实现对物理设备的高利用率。文献[7]基于蚁群算法,提出了一种有效的虚拟机放置策略。在同质和异构服务器的云环境中,将候选 VM 分组在一起,从全局优化的角度有效地减少了用于分配虚拟机的服务器数量,该策略适用于具有不同大小 VM 的各种分配问题。

考虑到云服务中计算成本、任务执行截止时间、服务质量(Quality of Service, QoS)、高吞吐量等约束条件,有效的任务调度策略成为云计算中一个重要的研究领域。为了解决云环境中的任务调度和资源分配问题, Yang 等人^[8]提出了一种基于 PSO 的算法,将每个子任务分配给适当的资源,并对资源上的子任务进行排序以实现此方案的目标。Verma 等^[9]提出基于双准则优先的粒子群算法(Bi-Criteria Priority based Particle Swarm Optimization, BPSO),在可用的云资源上调度 workflow 任务,并在约束下的执行时间最大限度地降低执行成本。Pandey 等人^[10]提出了另一种基于 PSO 的启发式方案,该方案基于 PSO 给出的解决方案优化了任务-资源映射的成本,兼顾了计算成本和数据传输成本。针对数据密集型工作流的高能耗问题,肖鹏等人^[11]提出通过引入“虚拟数据访问节点”的方法来量化评估 workflow 任务的数据访问能耗开销,并在此基础上设计了一种“最小能耗路径”的启发式策略。这些方法从不同方面对云计算中任务分配问题提出相应的优化方案,从而达到降低

能耗开销、提高系统能效。

目前用于解决云计算中任务分配问题的方法各种各样,其中任务分配旨在将适当的任务或作业映射到最合适的可用资源上,可将其抽象归纳为与云计算硬件设施效率有关的组合优化问题之一^[12]。多处理器调度是一个非常难的问题,为了解决这些问题,粒子群算法被广泛采用,各种应用研究的成果不断涌现。Rodriguez 等^[13]提出了一种基于元启发式优化技术的粒子群优化算法(Particle Swarm Optimization, PSO),该算法的目的是在满足期限约束条件的同时最小化总体 workflow 执行成本。文献[12]提出了一个云环境的多目标任务调度方案,尽量减少任务执行时间、任务传输时间、任务执行成本和提高 QoS。文献[14]提出了一个改进的云环境任务调度算法,结合了模拟退火(Simulated Annealing, SA)算法和粒子群算法(PSO)来提高资源利用率。Patel 等人^[15]提出了一种基于群体智能算法的混合蚁群优化(Ant Colony Optimization, ACO)/PSO 技术来优化多播树。有一些研究者对文献[16]中提出的基于粒子群优化(PSO)的任务和 workflow 调度方案进行了深入的分析,提出动态自适应粒子群优化算法(Dynamic Adaptive Particle Swarm Optimization, DAPSO)来增强基本 PSO 算法的性能,通过最小化特定任务集的完成时间来优化任务运行时间,同时最大化资源利用率^[17]。这些文献考虑到 PSO 算法没有交叉和变异运算,结构简单,相比其他算法简单易实现的特点,但缺乏对算法中惯性权重的动态调节,这会使系统易陷入局部最优,导致收敛精度低和不易收敛问题。

PSO 算法的搜索过程是一个通过迭代使搜索空间不断缩小的过程,其中全局搜索能力和局部搜索能力的平衡对算法的效率起着至关重要的作用^[18]。本文针对 PSO 收敛速度慢,易陷入局部最优的问题,提出一种改进的优化调度策略。考虑到粒子群算法中的惯性权重是影响搜索结果和收敛速度的关键因素^[19],本文提出了一种适用于云计算资源调度的 M-PSO 算法,来提高收敛速度以及搜索的良好适应性,从而减少系统开销。

1 系统模型

1.1 能耗模型

任务分配问题就是将应用程序的一系列任务合理分配到分布式计算系统中相应的可用资源上,以达到计算资源的负载均衡^[20],减少任务执行的等待

时间并提高系统输出效率. 用户对调度结果的评价依据有很多, 文中主要研究如何减少任务的完成时间.

本文在任务分配问题中, 考虑任务的执行能耗和任务间交互能耗. 结合能耗的计算公式 $E = P \times T$, 任务从进入云计算系统的时刻 t_0 到执行完毕时刻 t_1 所产生的能耗可表示为:

$$E(M) = \int_{t_0}^{t_1} P \cdot dt \quad (1)$$

定义. 云计算系统中任务间的交互关系可定义为任务交互图(Task Interaction Graph, TIG), 其中代表一个程序的任务, 并且 $T = \{1, 2, \dots, m\}$ 代表这些任务之间的交互. 边权重 $e_{ij} = (T_i, T_j)$ 表示在节点 i 和 j 之间的任务信息交互大小; 节点权重 w_i 对应于任务本身大小. 图 1 表示了一个任务交互图(TIG)的实例.

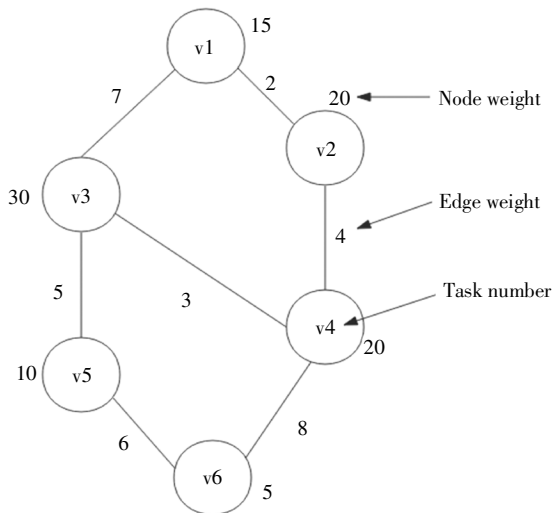


图 1 处理器上任务的 TIG 实例
Fig.1 The instance of TIG on processor

同样, 计算系统可由处理器交互图(Processor Interaction Graph, PIG) 表示为 $G(P, E)$, 其中 $P = \{1, 2, \dots, n\}$ 表示系统中的处理器集合和边权重 d_{jk} 在任何两个节点之间 j 和 k 表示相应处理器之间的路径长度. 分配问题可以正式表示如下:

$M: T \rightarrow P$ 其中 M 是任务集合 T 映射到处理器集合的函数. 该问题可以描述为: 找到一个任务处理器映射实例 M ; 使得在估计使用每个计算资源所引起的总成本 P 时, 所有计算资源中的最大成本被最小化. $E_{ex}(M)_k$ 是分配给计算资源 k 的所有任务的总开销. $E_{tr}(M)_i$ 是任务 i 与其他未分配在相同处理器上的任务之间的总交互开销. 在一个给定的任务分配

映射中, 系统总开销 $E(M)$ 是执行开销和交互开销的总和:

$$E(M) = \sum_{k=1}^n E_{ex}(M)_k + \sum_{i=1}^m E_{tr}(M)_i = \sum_{k=1}^n \sum_{i=1}^m \left(p_k \cdot \frac{w_i}{c_k} \right) + \sum_{i=1}^m \sum_{j=1}^m \left(p_k \cdot \frac{e_{ij}}{b} \right) \quad (2)$$

在估计所有资源的总开销时, 任务分配问题的目标是找到一个任务分配映射 M , 对于给定的 PIG 和 TIG 上具有最小的开销:

$$\text{Minimize}(E(M) \quad \forall M) \quad (3)$$

1.2 任务分配模型

在云环境中, 大量的云用户在不了解系统基础结构的情况下向云服务提供商提交独立任务并从云访问服务. 从任务的长度和任务的资源需求来看, 任务可视为是异构的^[2].

在本文中, 定义 m 维搜索空间对应于 m 个任务的分配问题. 每个维度都有随机的可能值集合 $s = \{x | 1 \leq x \leq n\}$, n 是系统中的处理器数量. 如: $\{(1, 2)(2, 3)(3, 4)(4, 1)(5, 2)(6, 3)(7, 5)\}$ (任务, 处理器) \dots 映射 \Rightarrow $\begin{pmatrix} \text{dimension:} & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ \text{value} & : & 2 & 3 & 4 & 1 & 2 & 3 & 5 \end{pmatrix}$ (一个 PSO 粒子) 表示在 PSO 域中的一个可能的赋值实例与粒子位置坐标之间的映射, 图 2 表示了一个任务分配模型. 使用这样的粒子表示法, PSO 总体被表示为由 $m \times n$ 二维数组组成的 n 粒子, 每个粒子表示 m 个任务的一个矢量. 因此, 一个粒子在 m 维搜索空间中飞行.

在 PSO 算法中, 我们将 m 个任务分配实例视为相关的 n 粒子坐标位置. 该算法首先随机产生大量的潜在分配方案作为粒子群的初始种群, 然后通过计算粒子的适应度函数值来更新粒子的位置.

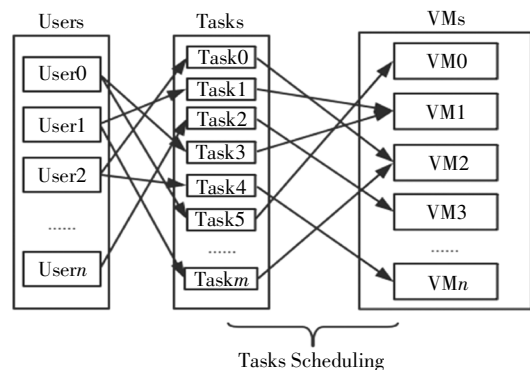


图 2 系统任务分配模型
Fig.2 The system task scheduling model

2 基于 PSO 的任务调度

本文讨论了以下情况下的任务分配问题. 该系统由一组具有各种存储器和处理资源的处理器组成, 这表明在不同处理器上执行的任务具有不同的执行成本^[22]. 任务将利用来自其占用的处理器的资源. 为了实现负载平衡, 以最大限度地提高处理器利用率, 同时最大限度地减少总任务执行时间, 负载均衡机制应该足够公平地分配处理器之间的负载. 定义如下适应度函数来评估粒子位置:

$$\text{fitness}(i) = \frac{1}{E(M)_i} \quad (4)$$

满足:

$$1 \leq i \leq p$$

p 是 PSO 种群的规模, 适应度函数评估粒子飞行期间位置的表现, 目标函数计算分配给处理器的一组任务的总成本. 每个粒子都有一个适应值, 并在每一次迭代中进行优化.

2.1 粒子群优化

粒子群优化算法是一种随机优化技术, 由鸟群或鱼群等社会行为原理产生. 在 PSO 系统中, 称为粒子的群体通过搜索空间飞行. 每个粒子代表优化问题的候选解决方案. 粒子的位置受到自身位置和访问其附近的最佳粒子位置来更新自己新的位置:

$$v_{id} = \omega v_{id} + c_1 R_1 (p_{id} - x_{id}) + c_2 R_2 (p_{gd} - x_{id}) \quad (5)$$

$$x_{id} = x_{id} + v_{id} \quad (6)$$

式中: v_{id} 为粒子的当前位置; ω 为惯性权重; x_{id} 为粒子的当前速度; p_{id} 为局部最优的粒子位置; p_{gd} 为全局最优的粒子位置; c_i 为加速系数; R_i 为取值在 $[0, 1]$ 之间的随机函数.

2.2 改进的任务分配 PSO 算法

在粒子群算法中有一个至关重要的元素是惯性权重, 它是影响搜索结果和收敛速度的关键因素. 为了克服 PSO 的收敛速度慢和局部最优的问题, 构建非线性递减的惯性权重函数将是有效的方法. 当惯性权重较大时, 粒子群优化算法具有较好的全局搜索能力, 可以在整体解中确定具体函数值的解; 当惯性权重值较小时, 粒子群算法具有较好的局部搜索能力, 可以使算法在较小范围内进行详细的局部搜索. 在粒子群优化的全过程中, 首先设置大的惯性权重, 并快速确定全局范围内的最优值的范围; 然后, 在算法后期设置较小的惯性权重, 使算法搜索最优值, 从而使搜索算法具有更好的搜索结果和更快的

收敛性. 构建如下惯性权重函数:

$$\omega(x) = \frac{a \cdot n}{x\gamma + n} + \beta \quad (7)$$

为了证明这个函数的有效性, 需要对它进行二次求导:

$$\omega(x)' = \frac{-\alpha\gamma n x^{\gamma-1}}{(x^\gamma + n)^2} \quad (8)$$

$$\omega(x)'' = \frac{\alpha\gamma n x^{\gamma-2}(x^\gamma + n)[(\gamma+1)x^\gamma - (\gamma+1)n]}{(x^\gamma + n)^4} \quad (9)$$

满足:

$$\begin{cases} \omega(x)'' < 0, x > \sqrt[\gamma]{\frac{(r-1)n}{r+1}} \\ \omega(x)'' = 0, x = \sqrt[\gamma]{\frac{(r-1)n}{r+1}} \\ \omega(x)'' > 0, x < \sqrt[\gamma]{\frac{(r-1)n}{r+1}} \end{cases}$$

由公式(8)可知, 一次导数小于零, 函数值随值的增大而减小; 由公式(9)可知, 二次导数存在零点, 函数有极值点, 函数值随 x 值非线性递减.

在 PSO 中, 惯性权重的范围是在 0.4 到 1.2 之间. 较大的 ω 值适合在新的全局空间搜索, 而较小的 ω 值适合搜索现有的局部空间. 要在局部搜索和全局搜索之间取得平衡, 则应该适当选择惯性权重因子. 如公式(9)所示, n 是算法迭代次数, x 代表当前的迭代次数, α 和 β 是阈值参数控制 ω 的范围, γ 是控制 ω 变化速度的加速因子. 图 3 显示了不同 γ 值的权重函数:

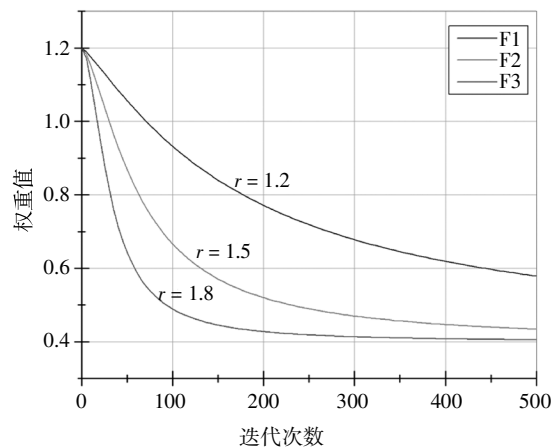


图 3 不同加速因子的惯性权重函数

Fig.3 Different acceleration factors of inertia weight function

PSO 算法在早期搜索时收敛速度非常快, 尽快进入局部搜索从而获得更好的解决效率. 从图 3 可以看出, 惯性权重函数是一个凹函数, 随着迭代次数的增加, 下降速度逐渐减慢. 因此, 在不影响收敛精

度的前提下,可以大大提高粒子群算法的收敛速度.

算法伪代码总结了所提出的任务分配问题的 PSO 算法:

Algorithm:

输入: 随机初始化粒子 PSO[i]

输出: 全局最优分配 G_{best}

步骤:

1)初始化

计算所有计算资源中所有任务的平均计算开销;

计算资源之间的平均传输开销;

初始化任务节点权重 w_i ;

初始化边权重 e_{ij} ;

计算 PSO(T_i);

2)While 存在未分配的方案 do

{ For $T_i \in T_i$ do

 分配任务 $\{T_i\}$ 到资源 $\{P_j\}$;

 更新每个粒子的 fitness 值;

 If fitness [G_{best}] \geq fitness [i]

$P_{best}[i] = PSO[i]$;

 计算 PSO($\{T_i\}$);

}

3 实验结果和讨论

3.1 实验设置

CloudSim 工具包是由 Gridbus 项目组 and 澳大利亚墨尔本大学网络实验室开发的开源模拟器, 本文使用 CloudSim 平台模拟云环境下的任务调度算法. 为了检验所提出的 M-PSO 调度算法的性能, 本文分别与改进粒子群优化算法(IPSO), 蚁群优化算法(ACO)混合 PSO-ACO 算法^[23]进行了比较. 实验参数如表 1 所示, 为了有效地评估这些调度算法, 本文分别在 20 个虚拟计算资源和 200, 300, 500 个任务的条件下进行实验. 在这些实验中, 每个 CPU 的处理能力分别是 1 000 MIPS, 1 500 MIPS 和 2 000 MIPS. 任务长度在[400 MI, 500 MI]的范围内.

另外在算法中使用了以下参数值:

1)迭代次数在[20, 200]范围以 20 为单位递增.

2)选择的加速因子取值在[1.0, 2.0].

3) $c_1 = c_2 = 1.0$.

4) α 和 β 分别设定为 0.8 和 0.4.

表 1 实验参数

Tab.1 Experimental parameters

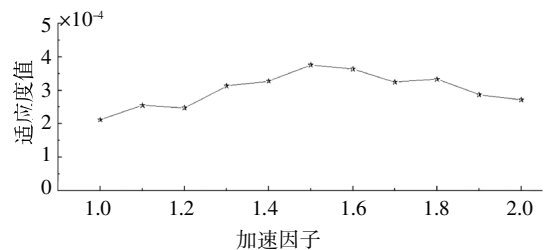
参数名称	参数值
Number of host	5
CPU(MIPS)	{1 000, 1 200, 1 500, 1 800, 2 000}
Memory size/ (MB)	2 048
Number of VM	30
Max no. iterations	320
Population size	40
Number of tasks	{200, 400, 600, 800, 1 000}

3.2 实验结果

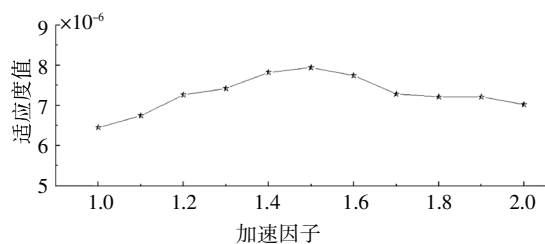
为了评估 M-PSO, IPSO, ACO 和 PSO-ACO 四种算法的性能, 比较了不同数据大小的系统开销, 并从不同角度对算法中各参数变量进行分析. 分别讨论了加速因子对算法适应度值的影响, 迭代次数对任务分配完成保证率的影响.

3.2.1 加速因子对适应度的影响

图 4(a)表示在任务量 300、迭代次数 100 时, 加速因子变量值从 1 逐渐递增到 2 的过程中对算法适应度值的影响. 由图可以看出, 适应度值变化的整体趋势为先增加再递减, 但变化过程中并非严格递增和递减, 其中存在轻微波动. 当加速因子值为 1.5 时, 对应的适应度值达到最大, 此时为算法的最优状态. 图 4(b)表示将任务量增加至 800, 算法迭代次数增加到 200 的情况下, 加速因子对适应度值的影响. 从图可以看出, 适应度值的变化趋势和图 4(a)相似, 同样呈现出先增后减的趋势, 并当加速因子取值在 1.4 至 1.6 之间达到峰值. 由此可以分析出, 在不同任务量和迭代次数下, 选取加速因子值为 1.5 时, 算法可以有较好的性能表现, 故在后面的对比实验中, 将其取值为 1.5 作为系统参数值.



(a)迭代次数 100, 任务量 300



(b)迭代次数 200, 任务量 800

图 4 不同条件下加速因子对适应度值的影响

Fig.4 Effect of acceleration factor on fitness value

3.2.2 迭代次数对保证率的影响

图 5(a)和 5(b)分别表示在加速因子变量取值为 1.5、任务量为 300 和 1 000 的条件下,迭代次数对完成任务分配的保证率的影响.图 5(a)中,任务量较少时,保证率随迭代次数的增加变化明显,且曲线的变化较为平滑.初期当迭代次数取值在 20 至 150 之间时,保证率上升速度快,说明此时算法对迭代次数要求比较高,适当增加迭代次数对系统有较高的保证率.但当迭代次数逐渐增加至 200 时,保证率曲线逐渐出现收敛现象,达到 250 时变化不明显.图 5(b)中,相应增加任务量,此时 IPSO 和 PSO-ACO 算法的保证率曲线变化十分相似,迭代次数在 50 至 200 区间时对应的保证率几乎一致,后期到达收敛时,IPSO 算法的保证率略微高于 PSO-ACO 算法.从图 5(a)和图 5(b)中可以看出,M-PSO 算法的保证率始终高于 IPSO 和 PSO-ACO 两种算法,并且三种算法在 200 次迭代以后保证率的变化都逐渐开始收敛,此时迭代次数对算法的影响较小.但随着任务量的增加,各算法本身的保证率也会随着略微降低.

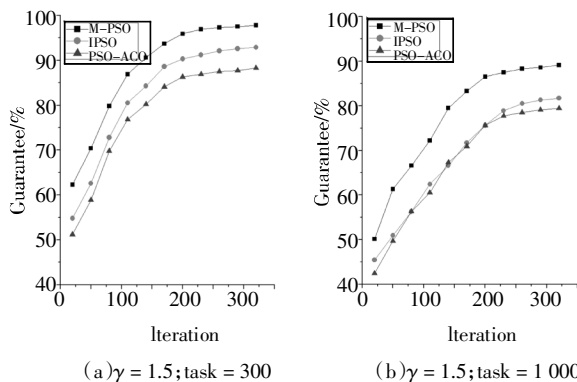


图 5 不同条件下迭代次数对保证率的影响

Fig.5 Effect of iteration on guarantee under different conditions

3.2.3 任务负载对系统能耗的影响

图 6 表示在加速因子取值为 1.5、迭代次数取值为 200 的条件下,不同的任务数所产生的系统开销.由图可以看出,在任务量为 200 时,算法 ACO 的系统开销处于最低,而 ACO-PSO 和 M-PSO 算法其次,IPSO 算法最高.其原因主要在于,与其他启发式算法相比,蚁群算法具有较强的鲁棒性和较好的求解性能.同时,在工作任务量较小的情况下,算法的迭代次数对基于 PSO 的各种改进算法影响不明显.当执行任务量为 400 时,算法 M-PSO 和 PSO-ACO 的开销较低,IPSO 算法和 ACO 算法次之,其原因主要为:当主机负载增加时,基于 PSO 的算法逐渐体

现出其在调度方面的优势.PSO-ACO 将 ACO 的局部搜索和变异操作同时混合到 PSO 算法中,通过适当调整各自的优势,从而使得 PSO-ACO 提高了全局优化的能力.当负载的任务量进一步增大时,M-PSO 算法的优化比率很明显.与 IPSO,PSO-ACO 和 ACO 相比,在 600 个任务和 200 次迭代的情况下,所提出的算法 M-PSO 分别降低了总开销的 8.5%,7.1%和 9.7%,而 PSO-ACO 与 IPSO 和 ACO 相比只降低了 2.9%和 5.7%的开销.

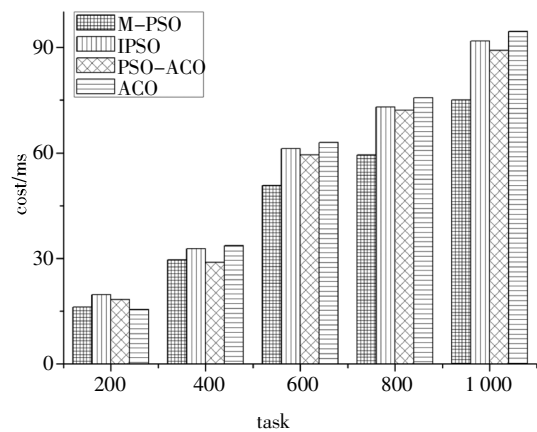


图 6 不同任务数对系统开销的影响

Fig.6 Effect of different number of tasks on system cost

后期任务量继续增加时,M-PSO 算法依然保持着较低开销的优势.由于 ACO 算法是一种典型的概率算法,算法中设置的参数通常由实验方法经验确定,导致性能的优化与人的经验密切相关,所以很难做到在不同任务量的情况下一直处于最佳的执行效果.而混合算法 PSO-ACO 的收敛速度较慢,随着任务量的不断增大算法容易陷入局部最优,导致算法需要较长的搜索时间,其复杂度可以反映出这一点. IPSO 算法最主要的问题在于容易过早收敛,这导致其丧失了搜索空间中的种群多样性,进行全局搜索的优化能力较差.

综合以上实验分析,随着负载任务量的不断增大,各算法的性能表现也有所变化.粒子群算法在处理大任务量时具有明显优势,并且加速因子、惯性系数和迭代次数对粒子群算法的性能表现具有较大影响.这意味着基于 PSO 算法的 M-PSO 算法通过对惯性权重系数的改进,克服算法收敛和局部最优问题能够实现以相对较低的系统开销执行较大的任务负载量.因此,在云计算环境中分配较大规模任务负载应用时,采用 M-PSO 算法具有更大的优势.

4 结论

本文提出了一种基于任务调度的改进粒子群优化算法(M-PSO),实验结果表明:1)在总开销方面,M-PSO算法在处理任务负载量较大时的性能优于负载量较小的;2)随着任务量和迭代次数的增加,所提出的M-PSO任务调度算法在降低总开销方面比其他3种算法(IPSO,PSO-ACO和ACO算法)更有效。该算法有望应用于实际的云平台,旨在提高效率,降低任务调度的总开销。

参考文献

- [1] ZENG L, VEERAVALLI B, ZOMAYA A Y. An integrated task computation and data management scheduling strategy for workflow applications in cloud environments [J]. *Journal of Network & Computer Applications*, 2015, 50(C): 39—48.
- [2] SINGH S, SWAROOP A, KUMAR A, *et al.* A survey on techniques to achieve energy efficiency in cloud computing [C]// *International Conference on Computing, Communication and Automation*. IEEE, 2017: 1281—1285.
- [3] LEBEL L, TRI N H, SAENGNOREE A, *et al.* Industrial transformation and shrimp aquaculture in Thailand and Vietnam; Pathways to ecological, social, and economic sustainability? [J]. *Ambio A Journal of the Human Environment*, 2017, 31(4): 311—323.
- [4] 罗亮, 吴文峻, 张飞. 面向云计算数据中心的能耗建模方法[J]. *软件学报*, 2014, 25(7): 1371—1387.
LUO L, WU W J, ZHANG F. Energy modeling based on cloud data center [J]. *Journal of Software*, 2014, 25(7): 1371—1387. (In Chinese)
- [5] ZHOU Z, ABAWAJY J, LI F M, *et al.* Fine-grained energy consumption model of servers based on task characteristics in cloud data center [J]. *IEEE Access*, 2018, 6(1): 27080—27090.
- [6] ZHOU Z, ABAWAJY J, CHOWDHURY M, *et al.* Minimizing SLA violation and power consumption in cloud data centers using adaptive energy-aware algorithms [J]. *Future Generation Computer Systems (FGCS)*, 2018, 86(6): 836—850.
- [7] LIU X F, ZHAN Z H, DENG J D, *et al.* An energy efficient ant colony system for virtual machine placement in cloud computing [J]. *IEEE Transactions on Evolutionary Computation*, 2018, 22(1): 113—128.
- [8] YANG Z, QIN X, LI W, *et al.* Optimized task scheduling and resource allocation in cloud computing using PSO based fitness function [J]. *Information Technology Journal*, 2013, 12(23): 7090—7095.
- [9] VERMA A, KAUSHAL S. Bi-criteria priority based particle swarm optimization workflow scheduling algorithm for cloud [C]// *Engineering and Computational Sciences*. IEEE, 2014: 1—6.
- [10] PANDEY S, WU L, GURU S M, *et al.* A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments [C]// *IEEE International Conference on Advanced Information NETWORKING and Applications*. IEEE, 2010: 400—407.
- [11] 肖鹏, 胡志刚, 屈喜龙. 面向数据密集型工作流的能耗感知调度策略[J]. *通信学报*, 2015, 36(1): 149—158.
XIAO P, HU Z G, QU X L. Energy-aware scheduling policy for data-intensive workflow [J]. *Journal on Communications*, 2015, 36(1): 149—158. (In Chinese)
- [12] RAMEZANI F, LU J, HUSSAIN F. Task scheduling optimization in cloud computing applying multi-objective particle swarm optimization [C]// *International Conference on Service-Oriented Computing*. Springer Berlin Heidelberg, 2013: 237—251.
- [13] RODRIGUEZ M A, BUYYA R. Deadline based resource provisioning and scheduling algorithm for scientific workflows on clouds [J]. *IEEE Transactions on Cloud Computing*, 2014, 2(2): 222—235.
- [14] KAUR G, SHAMA E S. Optimized utilization of resources using improved particle swarm optimization based task scheduling algorithms in cloud computing [J]. *International Journal of Emerging Technology in Advanced Engineering*, 2014, 4(6): 110—115.
- [15] PATEL M K, KABAT M R, TRIPATHY C R. A hybrid ACO/PSO based algorithm for QoS multicast routing problem [J]. *Ain Shams Engineering Journal*, 2014, 5(1): 113—120.
- [16] MASDARI M, SALEHI F, JALALI M, *et al.* A survey of PSO-based scheduling algorithms in cloud computing [J]. *Journal of Network & Systems Management*, 2017, 25(1): 122—158.
- [17] AL-MAAMARI A, OMARA F A. Task scheduling using PSO algorithm in cloud computing environments [J]. *International Journal of Grid and Distributed Computing*, 2015, 8(5): 245—256.
- [18] ROSTAMI A, LASHKARI M. Extended PSO algorithm for improvement problems K-means clustering algorithm [J]. *International Journal of Managing Information Technology*, 2014, 6(3): 17—29.
- [19] BOROWSKA B. Exponential inertia weight in particle swarm optimization [C]// *Information Systems Architecture and Technology: Proceedings of 37th International Conference on Information Systems Architecture and Technology C ISAT 2016 C Part IV*. Springer International Publishing, 2017: 265—275.
- [20] BHARATHI P D, PRAKASH P, KIRAN M V K. Energy efficient strategy for task allocation and VM placement in cloud environment [C]// *Power and Advanced Computing Technologies*. IEEE, 2018: 1—6.
- [21] MISHRA S K, PUTHAL D, SAHOO B, *et al.* An adaptive task allocation technique for green cloud computing [J]. *Journal of Supercomputing*, 2018, 74(1): 1—16.
- [22] 束束, 梁昌勇, 徐健. 基于信任的云服务系统多目标任务分配模型[J]. *计算机研究与发展*, 2018, 55(6): 1167—1179.
SHU J, LIANG C Y, XU J. Trust-based multi-objectives task assignment model in cloud service system [J]. *Journal of Computer Research and Development*, 2018, 55(6): 1167—1179. (In Chinese)
- [23] JU J H, BAO W Z, WANG Z Y, *et al.* Research for the task scheduling algorithm optimization based on hybrid PSO and ACO for cloud computing [J]. *International Journal of Grid & Distributed Computing*, 2014, 7(25): 217—218.