软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2016,27(9):2414−2425 [doi: 10.13328/j.cnki.jos.004849]

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

支持绿色异构计算的能效感知调度模型与算法*

王静莲[1,2], 龚 斌[2,3], 刘 弘[4], 李少辉[4]

[1](鲁东大学 信息与电气工程学院,山东 烟台 264025)

[2](山东大学 计算机科学与技术学院,山东 济南 250101)

[3](山东省高性能计算中心,山东 济南 250101)

[4](山东师范大学 信息科学与工程学院,山东 济南 250014)

通讯作者: 王静莲, E-mail: wjljing@163.com

摘 要: 异构调度可使大规模计算系统采用并行方式聚合广域分布的各种资源以提高性能.传统调度目标追求高性能而忽视高效能,远不能适应绿色计算科学发展新要求.因此,在理论上,一方面基于对动态频率和电压等系统参数的精细表述及有效量化,建立面向协同异构计算且易于复用的能效感知云调度模型;另一方面,提出并实现适于超计算机混合体系的多学科背景的元启发式多目标全局优化算法.从技术上解决了面向不同环境目标的云调度实施条件界定及其调度指标(时间、能效)实时变化描述等问题.大量仿真实验结果表明:与 3 个元启发式云调度器相比,该方法在能效及可扩展等方面优势明显;对于高维实例,整体性能改善分别达到 8%,12%和 14%.

关键词: 异构调度;绿色计算;协同进化;混合并行

中图法分类号: TP316

中文引用格式: 王静莲,龚斌,刘弘,李少辉.支持绿色异构计算的能效感知调度模型与算法.软件学报,2016,27(9):2414−2425. http://www.jos.org.cn/1000-9825/4849.htm

英文引用格式: Wang JL, Gong B, Liu H, Li SH. Model and algorithm of energy-efficiency aware scheduling for green heterogeneous computing. Ruan Jian Xue Bao/Journal of Software, 2016,27(9):2414−2425 (in Chinese). http://www.jos.org.cn/1000-9825/4849.htm

Model and Algorithm of Energy-Efficiency Aware Scheduling for Green Heterogeneous Computing

WANG Jing-Lian[1,2], GONG Bin[2,3], LIU Hong[4], LI Shao-Hui[4]

[1](School of Information and Electrical Engineering, Ludong University, Yantai 264025, China)

[2](School of Computer Science and Technology, Shandong University, Ji'nan 250101, China)

[3](Shandong High Performance Computing Center, Ji'nan 250101, China)

[4](School of Information Science and Engineering, Shandong Normal University, Ji'nan 250014, China)

Abstract: Designed to provide pervasive access to distributed resources in parallel ways, heterogeneous scheduling is extensively applied in large-scale computing system for its high performance. Conventional real-time scheduling algorithms, however, often overlook energy-efficiency while focusing on stringent timing constraints. To engage in green heterogeneous computing, a reusable energy-aware cloud model is first presented via mathematical formulation and quantization of the system parameters such as dynamic voltage and frequency scaling (DVFS), and dynamic power management (DPM). In addition, multidisciplinary context for multi-objective global optimization meta-heuristic is proposed and accomplished based on the supercomputer hybrid architecture. Furthermore, some technological breakthroughs are achieved with respect to boundary conditions for different heterogeneous computing and cloud

* 基金项目: 国家自然科学基金(61070017, 61272094); 国家高技术研究发展计划(863)(2006AA01A113, 2012AA01A306)

Foundation item: National Natural Science Foundation of China (61070017, 61272094); National High-Tech R&D Program of China (863) (2006AA01A113, 2012AA01A306)

收稿时间: 2015-01-22; 修改时间: 2015-03-23, 2015-04-14; 采用时间: 2015-04-22

scheduling, and descriptions of real-time variation of scheduling indexes (stringent timing constraints and energy-efficiency). Extensive simulation experiments highlight higher efficacy and better scalability for the proposed approaches compared with the other three meta-heuristics; the overall improvements achieve 8%, 12% and 14% for high-dimension instances.

Key words: heterogeneous scheduling; green computing; co-evolution; hierarchical parallelization

目前,大规模计算系统采用并行技术可具高吞吐信息服务和海量数据处理能力,在科学计算和金融等领域需求迅猛增长[1].随着规模的不断扩大,异构计算聚合广域分布的各种同构与异构的计算机、工作站、机群、群集、数据库、高级仪器和存储设备等资源,可形成对用户相对透明的、虚拟的高性能环境[2].并行系统效能的高低,很大程度上由部署在体系架构上的资源管理系统决定[3].任务调度是资源管理的核心,为了优化某个目标函数,其在一组具有任意特性的处理机中对任务集合进行排序和资源分配[4].当前,同构调度问题已被广泛研究[5];但异构调度,鉴于其复杂性、环境的多样性、应用的新需求和调度目标的折衷性等,是一个亟待解决的高维多模优化难题[6].

与此同时,绿色计算因为与环境保护和人类可持续发展的密切关联,引起越来越多的社会关注[7,8].2010 年,信息技术(information communication technology,简称 ICT)产业能耗占到总的全球电能消耗的 8%、全球能耗的 25%,成为全球第五大耗能产业;ICT 设备的 CO_2 排放占到全世界排放总量的 2%~2.5%,这个数字跟全世界航空运输业相当,并约等于汽车 CO_2 排放的 1/4.之后,ICT 总能耗逐年增长 5%,数据中心占其 40%.2011 年,中国数据中心能耗占全国能源耗电总量的 5%,其中,服务器占数据中心耗电量的 46%.以 2004 年落户上海超级计算中心的每秒 10 万亿次曙光 4000A 为例,这台机器每年光电费就要 400 多万元,平均每天就是 1 万多元.能源的消耗远远大于超级计算机自身的价值.与此同时,数据显示,ICT 产业存在能源浪费,其中,全世界数据中心目前的服务器能力平均利用率只有 10%~30%.因此,高性能领域的绿色计算成为数据和计算中心实际运行的关键问题[9].

另外,对高度数据密集型工作负载的支持,正成为下一代计算和数据中心的关键技术.这里,实时任务据其间的依赖性,分为独立任务或依赖任务应用;而数据密集型应用是指以数据为中心,存在海量数据传输的依赖任务应用.计算资源的异构性、调度技术的局限性和调度指标的平衡性,是面向数据密集应用调度研究所需考虑的重要因素[10].

再者,调度算法的研究主要集中在基于需求建模的启发式算法[11−14]和基于进化理论的元启发式算法[15−23].

对任务的多项需求,启发式调度会将多目标聚合成单一目标函数处理.这种方法简单而有效,因此被广泛研究和采用.但其自身也存在一些固有的缺陷:由于多目标优化问题的解并非唯一,而是存在一个最优解集合,称为非劣解;而单目标优化算法仅能根据聚合函数得到决策空间的一个可行解,降低了最终解的质量,缺乏灵活性和扩展性.

更为合理的途径是采用多目标组合最优化算法来解决这个问题,基于进化理论的元启发式算法是较为有效的方法.但面对其复杂多样性,目前算法大多存在两个瓶颈:种群的收敛速度较慢或个体多样性不能保持.并行与分布式元启发式算法(parallel distributed meta-heuristics,简称 PDM)因在较大目标空间的搜索高效性及鲁棒性,近年来被广泛应用,具体分为 4 类:主从模型、细粒度模型、粗粒度模型和混合模型.其中,粗粒度模型被广泛应用在超计算机体系结构上[24−26].但上述 PDM 在面向大规模实时实例时,运行算法的超计算机虽然能达到峰值需求,很多时候却效率不高.因此,面向多核 CPU+GPU 混合的超计算机体系结构,元启发式算法的并行设计,也是决定问题高效求解的重要元素之一.

本文在对大规模计算系统从高性能向高效能发展的充分需求调研、分析基础上,为解决当前异构资源管理及调度存在的问题,以适应不同环境及应用的新需求和新特征,围绕节能减排和调度协同等热点,重点展开云计算中融合能效感知调度建模及相关算法等方面的研究.本文主要研究工作和创新点如下.

(1) 综合分析当前异构调度技术的发展、局限和空白;

(2) 提出融合能效感知的云调度框架和模型;

(3) 设计多学科启发的协同进化全局多目标优化算法;

(4) 实现算法评测系统和仿真方法.

1 相关工作

调度问题研究如何把任务集或一组程序在时间和空间上最优化地分配到一组处理单元,以满足一定的调度目标,一般由 3 个部分构成:系统模型、应用模型和调度目标[8,9].近年来,异构调度模型除简单时限或基于成本约束以外,基于能耗约束[11−14]引起学者广泛关注.

能耗管理技术的研究包括两个方向:基于硬件体系结构的低功耗方法[10]和基于软件的功耗感知方法[11−14].前者通过低功耗的组件改变嵌入式系统架构,以减少能源消耗.这种方法是有效的,但它依赖于特定元件,具有设计和更换费用昂贵等缺陷;后者对系统性能和能耗指标之间实现折衷,能以更低的成本广泛可行.而功耗感知调度,采用操作过程中能耗监管和优化调节技术(例如动态电压频率缩放(dynamic voltage and frequency scaling,简称 DVFS)和动态电源管理(dynamic power management,简称 DPM)),属后者[11−14].

根据不同的调度应用,功耗感知调度一般分为独立任务和依赖任务调度.针对大规模计算,文献[11]研究同构集群优化调度算法,以求解时间和空间不同维度下共享资源的能耗优化;文献[12]提出云环境面向独立任务的能耗优化算法,即,基于博弈理论调节短时间作业的响应时间期望加快执行速度,其实质也是 DVFS 技术的使用;文献[13]实现同构云多作业的功耗感知调度,可借助休眠工作站减少能耗,并用 Pareto 最优前沿来折中性能和成本指标;文献[14]考虑的温度和热约束的面向独立任务的时限调度算法,允许云计算长时间运行作业通过在不同节点间的迁移来优化能耗值.然而,上述研究仍然有一些局限性,例如 DVFS 技术的不可实施性、静态能量重要性的忽略、数据密集型应用程序的不适合以及异构计算环境的不可扩展性等.

启发式算法是相对于最优化算法提出的,在可接受的花费(指计算时间和空间)下给出待解决组合优化问题每一个实例的一个可行解,该可行解与最优解的偏离程度不一定事先可以预计.对任务的多项 QoS 需求,启发式调度会将多目标聚合成单一目标函数处理.这种方法简单而有效,因此被广泛研究和采用,但其自身也存在一些固有的缺陷:由于多目标优化问题的解并非唯一,而是存在一个最优解集合,称为 Pareto 最优或非劣解;而单目标优化算法仅能根据聚合函数得到决策空间的一个可行解,降低了最终解的质量,缺乏灵活性和扩展性.

元启发式算法应用在异构调度问题已有十余年,包括遗传算法[15]、模因算法[16]、细胞模因算法[17]以及遗传算法与其他一些技术的融合,比如变邻域搜索[18]、神经网络[19]和列表调度技术[20].另外,文献[21]提出元启发式算法求解能耗感知调度问题,文献[22]研究了元启发式算法在多处理器优先约束的云调度问题中应用以及文献[23]主要探讨实时云调度的元启发式算法.但面对大规模云调度实时实例的复杂多样性,上述研究算法存在两个瓶颈:种群的收敛速度较慢或个体多样性不能保持.

元启发式算法的并行与分布式设计具体可划为 4 类:主从模型、细粒度模型、粗粒度模型和混合模型.其中,粗粒度模型也称为孤岛模型,被广泛应用在超计算机体系结构上.该模型将种群划分成若干个子群,子群各自独立进行相关进化操作,而子群间借助迁移算子通过个体交换的方式相互影响.随着多核 CPU+GPU 混合结构超计算机服务器的兴起,相关的几个孤岛模型研究主要集中在选择函数、替代函数、迁移率或拓扑结构的设计上.文献[25]面向超计算机计算平台,基于遗传算法和孤岛模型提出了新的并行进化算法 pCHC,用于求解云调度问题.该算法设计了跨世代精英选择、异构重组和大变异这 3 个算子,针对低维调度问题表现较好,但它存在两个瓶颈:结构化方案的搜索能力有限以及个体多样性保持不够.文献[26]在上述工作的基础上提出了 Pμ-CHC;即,在子群们并行进行相关进化操作的同时引入一个局部随机搜索策略.研究应用在经典问题实例和一些新的云调度实例上,效果较好.但上述研究在面向大规模实时实例进行云调度求解时,运行算法的超计算机虽然能处理高峰需求,很多时候却效率不高.

2 数学建模

能效感知云调度模型可以设计成一个中间件,然后插入到云资源系统中,从而使云任务能够有效分配到目标性能达到最优的节点上执行.这样一方面有效降低了云任务调度长度,另一方面,尽可能地提高了任务执行的完全性和可靠性,较好地满足了云应用的服务质量需求.云资源任务调度的体系架构如图 1 所示.

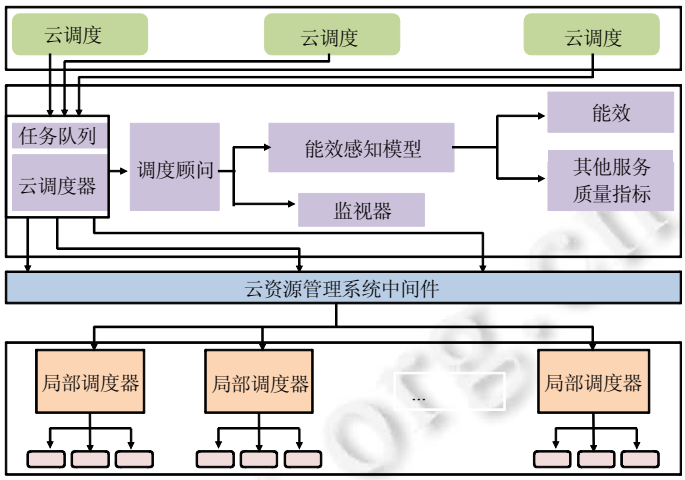

Fig.1 Cloud scheduling framework integrated with energy-efficiency awareness

图 1 能效感知的云调度体系架构

云资源任务调度体系架构的最底层为云资源层(包括节点局部调度器),第 2 层为云中间件层,第 3 层为任务全局调度器层,最上一层为云应用任务客户端.

在多目标优化云资源的任务调度框架下,云用户提交一个任务并被执行的具体过程如下:

1) 应用客户端将云应用提交到任务队列进行排队;
2) 任务调度器(全局)从任务队列中取出一个云应用中所有具有依赖关系的子任务,同时与调度顾问进行交互;
3) 调度顾问与云监视器及能效融合模型交互;
4) 云监视器将网络目前链路状态和可用资源信息反馈给调度顾问;
5) 模型通过分析云监视器反馈的可用资源信息,对资源节点的调度成功率及利用率进行分析获得行为能效性;同时,根据资源节点加密水平等获得固有安全性,然后将云资源的能效收益等信息反馈给调度顾问;
6) 调度顾问将云监视器和模型反馈的信息传递给任务全局调度器,通过云中间件将所有子任务分配到较优资源节点集执行;
7) 局部调度器对其控制范围内云资源节点上任务进行最优调度.

实时应用任务是由用户上传的若干子任务 $\alpha=\{\alpha_i\}(i=1,2,\ldots,m)$组成,具体可由一组属性集合表述:$\alpha_i=(b_i,\omega_i,\tau_i,v_i,\varsigma_i,l_i,S_i)$.其中,$b_i,\omega_i$ 及 τ_i 表示子任务的到达、估计执行和结束时间;v_i 表任务截止时间;ς_i 表计算量(周期数目);l_i 表示需保护的数据量(单位:KB);$S_i\rightarrow\kappa$(κ是一个正实数集)表示实时任务对云资源节点的安全需求集合,这里通常是指保密、诚信和认证这 3 种.

云资源节点集合可表述为 $\varphi=\{\varphi_j\}(j=1,2,\ldots,n)$.其中,每个节点 φ_j 有不同的属性值,见表 1.表 1 中第 1 列表示属性变量名称,第 2 列显示该参数是否是输入参数,第 3 列说明了参数的数据类型,最后一列详细表述了变量的代表意义.

通常,云调度模型中的主机 CPU 可拥有不同工作频率.对于指定频率 F_i,能耗模型定义为

$$P(F_i)=\alpha(P_f(F_i)-P_d(F_i))+P_d(F_i) \tag{1}$$

其中,α表示 CPU 负载,$P_f(F_i)$和 $P_d(F_i)$表示集群中所有主机单位时间内 100%CPU 和 0%CPU 分别采用频率 F_i 的功耗.由于集群中每台主机的能耗总量与其总体运行时间(ΔT_{h_i})成正比,因此能耗公式定义为

$$P_{(F)}=\sum_{i=1}^{n_H}P(F_i)^{h_i}\times\Delta T_{h_i} \tag{2}$$

Table 1 Model description and notation

表 1 模型属性参数

(属性)变量名称	是/否输入参数	类型	代表意义
F^h	是	集合	主机 h 所应用的 CPU 频率集合
n_F^h	是	整数	主机 h 所应用的 CPU 频率集合的元素个数
$\xi_{f,cpu}^{h,k}(t)$	否	整数	主机 h 在 t 时刻的 CPU 最大吞吐量
$\xi_{g,mem}^{h,k}(t)$	是	整数	主机 h 在 t 时刻的内存最大容量
$\omega_{f,cpu}^{h,k}(t)$	否	实数	主机 h 在 t 时刻的 CPU 占用比例
$\omega_{g,mem}^{h,k}(t)$	否	实数	主机 h 在 t 时刻的内存占用比例
V^h	否	集合	主机 h 上运行的虚拟机集合
$\omega_{f,cpu}^{v,h}(t)$	否	实数	主机 h 虚拟机 v 在 t 时刻的 CPU 占用比例
$\omega_{g,mem}^{v,h}(t)$	否	实数	主机 h 虚拟机 v 在 t 时刻的内存占用比例
n_V^h	否	整数	主机 h 上运行的虚拟机集合的元素个数
TY^h	是	整数	主机 h 类型(依据能耗的异构性)
$P_{\min}^{h,k}(F_i,Ty^h)$	是	整数	主机 h0%CPU 采用频率 F_i 的功耗
$P_{\max}^{h,k}(F_i,Ty^h)$	是	整数	主机 h100%CPU 采用频率 F_i 的功耗
H^k	是	集合	集群 k 的主机集合
n_H	是	整数	集群 k 主机集合的元素个数
$P^k(t)$	否	整数	集群 k 在 t 时刻的功耗

执行时间取决于要执行的需求任务复杂度.令虚拟机 v 待执行指令数目 κ^v,需求任务执行时虚拟机 v 的执行力为 $\omega_{*,*}^{v,h}$,则执行时间定义为

$$T=\max(\kappa^v/\omega_{f,cpu}^{v,h}) \tag{3}$$

鲁棒性可解释为一台主机发生故障所导致的失联虚拟机数目,即,每台运行主机的虚拟机平均数目,具体定义为

$$R=\sum_{i=0}^{n_H} n_V^{h_i}\Big/n_H \tag{4}$$

计算动力(或潜力)记作 D,代表现运行所有主机可继续利用的计算能力,其可引导云计算提供商在计算峰值来临之前暂不新增虚拟机数目.如果主机可具动态电压频率缩放,计算动力依据所有主机 CPU 的最大计算能力和当前计算力可定义为

$$D=\left(\sum_{i=0}^{n_H}\xi_{f,cpu}^{h_i,k}-\omega_{f,cpu}^{h_i,k}\right)\Big/n_H \tag{5}$$

其中,f 是 CPU 的最高频率,n_H 表正在运行的主机数目.

融合能效感知的云调度目标就是:在满足任务间依赖关系同时,寻找任务需求模型与资源拓扑结构之间的映射调度方案$\{(\alpha_i,\varphi_j)\}$,$i\in[1,m]$,$j\in[1,n]$,使云计算任务的调度长度和系统节点能效收益达到最优,求尽可能多的(甚是是全部)的非劣解,可以描述为

$$\text{Min } Y=F(X)=(f_1(X),f_2(X),\ldots,f_\kappa(X))=\{P_{(F)},T,R,1/D\} \tag{6}$$

其中,$X\in\Phi$,$Y\in\Omega$.X 称为决策变量,Φ是决策空间;Y 为目标函数值,Ω是目标函数空间.

3 多学科启发的协同进化多目标优化算法

在免疫学中,抗原是导致免疫系统产生抗体的物质.对于多目标优化问题,抗原被定义为目标函数(见公式(6)).

B 细胞、T 细胞及一些具抗原特异性的淋巴细胞通常称为抗体.人工免疫系统中抗体代表抗原的一个候选解.令 X 为抗体空间,抗体种群表示为 N-维抗体集合(N 是抗体种群规模);且抗体由基因组成,表示为 $x_i=(G_1,G_2,\ldots,G_N)$.在实际应用中,给定优化问题具有 N-维实数的目标搜索空间,一个候选解 $x_i(i\in[1,N])$由 N 维实数组成,

而每一维代表一个问题变量并被看作基因.这里,基因的基本单位是基因座.

认知心理学认为,模因是文化信息单位,是文化复制、传播和发展的基因.模因作为一种选择与建构的创新思维和科学方法,是社会进化原动力的一个表现形式.给定一个抗体 $A_\beta(A_\beta\in X)$,本文将抗体基因的实时进化信息看作模因,并形式化为 M.矩阵中每一维数据都与相应的基因进化信息对应,Z,N 表示模因空间的维数.

本文提出多学科启发的基因-模因协同进化调度算法(MCMC),即:首先,基于调度实际问题的离散空间特征进行抗体编码与解码;然后设计克隆、交叉、变异和选择等算子.算法框架如图 2 所示.MCMC 较人工免疫算法有 4 个方面改进:抗体基因亲和度值的细粒度评估、模因的数学表述、基因-模因协同进化过程的高效模拟以及结合孤岛模型和主从模型的多层次并行化设计.

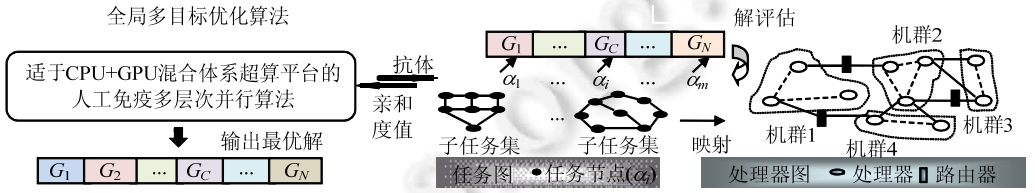

Fig.2 Algorithm architecture of global and multi-objective optimization for cloud scheduling problems

图 2 求解云调度问题的全局多目标优化算法框架

3.1 基因-模因协同进化模拟

克隆算子在算法中对解的多样性分布和逼近性起着重要的作用,通常,其对抗体种群 $\delta=\{\delta_1,\delta_2,\ldots,\delta_\varepsilon,\ldots,\delta_\theta\}$ 的操作 Γ_C 可以定义为式(7).

$$\begin{aligned}\delta^*(t)&=\Gamma_C\{\delta_1(t),\delta_2(t),...,\delta_\varepsilon(t),...,\delta_{\theta(t)}(t)\}\\&=\Gamma_C(\delta_1(t))+\Gamma_C(\delta_2(t))+...+\Gamma_C(\delta_\varepsilon(t))+...+\Gamma_C(\delta_{\theta(t)}(t))\\&=\{\delta_1^1(t),\delta_1^2(t),...,\delta_1^\varepsilon(t),...,\delta_1^{\ni 1}(t)\}+...+\{\delta_{\theta(t)}^1(t),\delta_{\theta(t)}^2(t),...,\delta_{\theta(t)}^\varepsilon(t),...,\delta_{\theta(t)}^{\ni\theta(t)}(t)\}\end{aligned}\tag{7}$$

这里,$\Gamma_C(\delta_i(t))=\{\delta_i^1(t),\delta_i^2(t),...,\delta_i^\varepsilon(t),...,\delta_i^{\ni i}(t)\}$,$i=1,2,\ldots,\theta(t)$;$\ni i\in[1,m_c]$是可调节参数,表示克隆概率;$m_c$ 是克隆概率上限;$\alpha\ni i=1$ 代表不对 $\delta_i(t)$进行克隆操作.由于多目标优化问题通常有非劣解集,因此通常对每个抗体 $\delta_i(t)$的克隆概率上限都可设定为 m_c.

在 MCMC 算法中,对每个抗体采用一致的克隆概率$\ni$,使得优化过程中的可行非支配解集规模几乎成倍上升,保持个体多样性同时还可加速群体收敛.具体如式(8)所示.

$$\delta^*(t)=\{\delta_1^1(t),\delta_1^2(t),...,\delta_1^\ni(t)\}+...+\{\delta_{\theta(t)}^1(t),\delta_{\theta(t)}^2(t),...,\delta_{\theta(t)}^\ni(t)\}\tag{8}$$

与此同时,在同一抗体的大量克隆个体上采用不同的基因交叉、变异策略,可有助于抗体间的协同和信息交换.这里,对抗体种群的基因操作 Γ_G 可定义为式(9).

$$\begin{aligned}\delta^{**}(t)&=\Gamma_G(\delta^*(t))\\&=\Gamma_G(\{\delta_1^1(t),\delta_1^2(t),...,\delta_1^\ni(t)\}+...+\{\delta_{\theta(t)}^1(t),\delta_{\theta(t)}^2(t),...,\delta_{\theta(t)}^\ni(t)\})\\&=\{\Gamma_G(\delta_1^1(t))+\Gamma_G(\delta_1^2(t))+...+\Gamma_G(\delta_1^\ni(t))\}+...+\{\Gamma_G(\delta_{\theta(t)}^1(t))+\Gamma_G(\delta_{\theta(t)}^2(t))+...+\Gamma_G(\delta_{\theta(t)}^\ni(t))\}\\&=\{\delta_1^{1\#}(t),\delta_1^{2\#}(t),...,\delta_1^{\ni\#}(t)\}+...+\{\delta_{\theta(t)}^{1\#}(t),\delta_{\theta(t)}^{2\#}(t),...,\delta_{\theta(t)}^{\ni\#}(t)\}\end{aligned}\tag{9}$$

其中,$\Gamma_G(\delta_i^j(t))=\delta_i^{j\#}(t)$ $(i=1,2,\ldots,\theta(t),j=1,2,\ldots,\ni)$表示抗体 $\delta_i^j(t)$ 以概率 P_m 进行的基因操作(包括交叉和变异).通常,工免疫系统模拟二进交叉算子或多项式变异算子.在 MCMC 算法中,交叉点或变异点的选取依据模因信息矩阵.

与克隆相反,选择操作将种群个体划分为非劣解或劣解,且只有非劣解可进入下一代.对于任何抗体 $\delta^\#(t)\in\delta^{**}(t)$,如果 $\delta^\#(t)$满足公式(10),为非劣解,否则是劣解;而对抗体种群的选择操作 Γ_S,定义为式(11).

$$\neg\exists\delta_\kappa^{\varpi\#}(t)\neq\delta^\#(t)(\kappa=1,2,...,\theta;\varpi=1,2,...,\ni)\in\delta^{**}(t):(\forall i\in\{1,...,m\}:f_i(\delta^\#(t))\geqslant f_i(\delta_\kappa^{\varpi\#}(t)))\tag{10}$$

$$\begin{aligned}\delta^{***}(\iota)&=\Gamma_S(\delta^{**}(\iota))\\&=\Gamma_S(\{\delta_1^{1\#}(\iota),\delta_1^{2\#}(\iota),...,\delta_1^{\ni\#}(\iota)\}+...+\{\delta_{\theta(\iota)}^{1\#}(\iota),\delta_{\theta(\iota)}^{2\#}(\iota),...,\delta_{\theta(\iota)}^{\ni\#}(\iota)\})\\&=\Gamma_S(\{\delta_1^{1\#}(\iota),\delta_1^{2\#}(\iota),...,\delta_1^{\ni\#}(\iota),...,\delta_{\theta(\iota)}^{1\#}(\iota),\delta_{\theta(\iota)}^{2\#}(\iota),...,\delta_{\theta(\iota)}^{\ni\#}(\iota)\})\\&=\{\delta_1^{\#}(\iota),\delta_2^{\#}(\iota),...,\delta_\varepsilon^{\#}(\iota),...,\delta_{\theta s(\iota)}^{\#}(\iota)\}\end{aligned}\tag{11}$$

其中, $\delta_\varepsilon^{\#}(\iota)$ (ε=1,…,$\theta s(\iota)$)表抗体群$\delta^{**}(\iota)$中的非劣个体,$\theta s(\iota)$是非劣个体的数目.通常,人工免疫系统根据当代每个个体的约束偏离值选取可行解集,然后,据每个个体的目标函数值选取可行非劣解集进行个体选择;而MCMC依据模因信息矩阵直接在种群中选取非劣抗体,有利于简化算法.

3.2 算法描述

MCMC 算法.

1: Initialize the iteration (ι) and the subpopulations $\{\delta_1(\iota),\delta_2(\iota),\ldots,\delta_\varepsilon(\iota),\ldots,\delta_{\theta(\iota)}(\iota)\}$, each of s individuals;
2: **While** ($\iota<\iota_{\max}$) and (other termination criteria are not satisfied)
3: **Do in parallel** for each island /*Obtain the coarse-grained model (also named as island model), one of the parallel and distributed models*/
4: $\iota=\iota+1$;
5: **Do in parallel** /*Obtain the master-slave model*/
6: Evaluate the affinity between the antibody and antigens (Eq.(6)) in the current population: $\Phi(A_\beta)(A_\beta\in\boldsymbol{X})$;
7: **For** (every couple of antibodies denoted as A_U and A_W)
8: **If** ($\Phi(A_U)>\Phi(A_W)$)
9: **For** (C=1; C<=N; C++)
10: **For** (K=1; K<=Z; K++)
11: **If** (A'_U G_{KC}}<> A'_W G_{KC})
12: Update A'_U meme vectors:
13: {$\boldsymbol{M}_{KC}(t)=(1-\rho)\times\boldsymbol{M}_{KC}(t-1)+\Delta\boldsymbol{M}_{KC}$;
14: $\Delta\boldsymbol{M}_{KC}=Q(t)/\Phi(A_U)$;}
15: Update A'_U other meme vectors $\boldsymbol{M}_{JC}(J<>K)$:$\boldsymbol{M}_{JC}(t)=(1-\rho)\times\boldsymbol{M}_{JC}(t-1)$;
16: **EndIf**
17: **EndFor**
18: **EndFor**
19: **EndIf**
20: **EndFor**
21: **End Do in parallel**
22: Perform clonal operation $\delta^{*}(\iota)=\Gamma_C\{\delta_1(\iota),\delta_2(\iota),\ldots,\delta_\varepsilon(\iota),\ldots,\delta_{\theta(\iota)}(\iota)\}$;
23: Perform gene operations based on meme matrices $\delta^{**}(\iota)=\Gamma_G(\delta^{*}(\iota))$;
24: Perform clonal selection operation $\delta^{***}(\iota)=\Gamma_S(\delta^{**}(\iota))$;
25: Save the best solution in the generation;
26: **If** $\iota=\tau$ (migration interval) then
27: Create Ψ_δ for the current subpopulation;
28: Send Ψ_δ to the neighboring subpopulation;
29: Receive Ψ_δ from the neighboring subpopulation;
30: Construct the founding subpopulation σ_δ;

31: Select s individuals in σ_δ;

32: Replace the subpopulation Ψ_δ with Ψ_δ^τ;

33: **End If**

34: **End Do in parallel**

35: **End While**

36: Output the best individual.

算法 MCMC 的第 7 步~第 20 步细粒度评估抗体基因的亲和度值,并有效模拟种群自组织的模因更新;而第 22 步~第 24 步定义了模因传播并影响基因进化的过程.

面向新近发展的混合多核 CPU+GPU 的高性能计算机集群体系结构,本文提出融合粗粒度模型和主从模型的层次并行模型.即:首先依据粗粒度模型将种群划分成若干子群,并把每个子群分配到一个节点上;而在每一个节点上,大量的个体适应度评估计算是适于 GPU 加速的主从式并行应用.这里,CPU 可看作主服务器,而在 GPU 多核上执行的若干线程就是客户端.第 26 步~第 33 步具体描述了多层次模型之一的粗粒度模型(也称为孤岛模型)采用的基于模因库的并行迁移策略.

3.3 算法的复杂度分析

设在每一代进化中,种群 *FeaNonPop* 和 *ModNonPop* 的规模都为 θ,克隆倍数为 $\ni$,变量的维数为 $£$,约束维数为 $\hbar$,目标函数维数为 m,则:

- 在每次克隆种群 *ModNonPop* 所用的复杂度为 $O(\ni\theta)$;
- 交叉操作所需复杂度为 $O(£\ni\theta/2)$;
- 变异操作所需复杂度为 $O(£\ni\theta)$;
- 计算种群 *Pop* 基因亲和度值的时间复杂度为 $O(£\ni\theta)$;
- 合并种群 *ModNonPop* 所需复杂度为 $O(£\theta+\hbar\theta+m\theta)$;
- 修正种群 *Pop* 中个体模因值所需复杂度为 $O(3m(\ni+1)\theta+2\hbar(\ni+1)\theta)$;
- 选取并更新可行非支配解集所需复杂度为 $O((2\ni+6+m\ni+2m)\theta+m(\ni+2)^2\theta^2+(\ni+2)(m+1)\theta\log_2((\ni+2)\theta))$;
- 选取并更新非支配解集所需复杂度为 $O((m+1)(\ni+1)\theta+\theta+m(\ni+1)^2\theta^2+(m+1)(\ni+1)\theta\log_2((\ni+1)\theta))$.

4 仿真实验及结果

实验在山东省高性能计算中心进行,采用浪潮天梭 TS10000 高性能集群系统,英特尔至强 5600 系列处理器(2.66GHz,12MB Cache),CPU+GPU 混合结构,共有 960 个计算核数,计算峰值达每秒 10 万亿次双精度浮点运算,内连 40Gb/s 带宽 1μs~2μs 超低延迟的高速网络.算法的并行实现采用 C-CUDA 编程语言和 MPICH-VMI(MPICH 1.2.7p1 版本)[27].表 2 总结了实验集群的相关参数设置.

Table 2 Simulator and simulation parameters

表 2 实验的相关参数设置

名称	参数值(Fixed)-(Varied)
CPU 速度	(100 million instructions/second or MIPS)-(100,200,…,800)
实时任务截止时间	(1000ms)-(1000,2000,…,100000)ms
节点数目	(64)-(8,16,32,64,96,128,256)
云资源节点 (Severs,F(GHz),P(W))	(IBM,2.13,675)(IBM,2.13,670)(IBM,2.66,1440)(IBM,2,1350) (IBM,1.86,1975)(IBM,2.33,310)(IBM,2.26,670)(IBM,3,400)(HP,2,460) (HP,2,750)(HP,2.4,460)(HP,2.4,300)(HP,2.4,920)(DELL,2.4,345)(DELL,2.4,305) (DELL,2.13,345)(DELL,2.26,1100) (DELL,2.33,345)

4.1 整体性能比较

文献[25]提出了一些新的模拟中型规模云调度实例集,并可以通过网站 http://www.fing.edu.uy/inco/grupos/

cecal/hpc/HCSP 下载应用.这些测试集是依据文献[28]所述的建模方法随机产生的,其目的是模拟复杂的异构计算环境.实例维数(任务×机器)包括 1024×32,2048×64,4096×128 及 8192×256,规模远大于文献[6]的经典的 12 个实例.

首先,针对高维异构依赖任务云调度问题,MCMC 按实例的一致性、半一致性及不一致性分类与算法 Min-Min[11]和 Sufferage[11]进行性能比较.这里,一致性、半一致性及不一致性的定义依从文献[25].

Min-Min 和 Sufferage 是能在合理时间内求解低维异构调度问题的两种较好启发式算法.从图 3 看:对于每维的一致性实例,算法 MCMC 相较 Min-Min 和 Sufferage 的时间性能改善约为 9%;而对半一致性实例,时间性能改善上升为 16%.另外,虽然对于低维的不一致性实例,MCMC 相较 Min-Min 和 Sufferage 的时间性能改善不明显,但面向高维的不一致性实例,其时间性能改善超过 15%.

然后,针对上述同样实例,MCMC 与目前面向计算机集群体系结构设计的较好的 3 种元启发式算法(IM-dDE-CUDA[24],pCHC[25]和 Pμ-CHC[26])进行性能比较.如图 4 所示:随着云调度问题的规模增大,MCMC 表现的性能改善越大.值得注意的是:对于高维实例 4096×128,MCMC 相较 Pμ-CHC,pCHC 和 IM-dDE-CUDA 的性能改善分别达到 8%,12%和 14%.

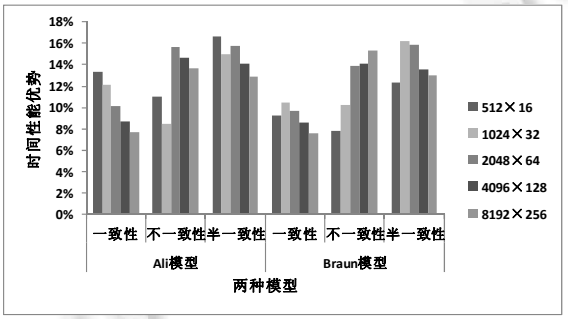

Fig.3 Time comparisons with deterministic heuristics

图 3 与确定性启发式算法的时间性能比较

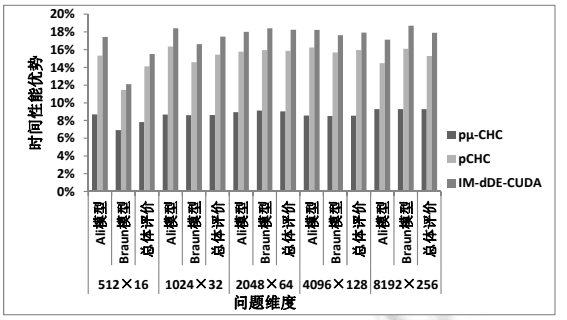

Fig.4 Time comparisons with the meta-heuristics

图 4 与元启发式算法的时间性能比较

4.2 融合能效感知模型的影响

数据中心内共有 N=200 台服务器,通常,每台服务器都有淘汰周期(一般为 3 年).可以假定服务器的最优值取决于机器的使用时长.简单起见,我们假设 1/3 的服务器的最优点为(0.8,0.9),即:其 CPU 和硬盘利用率分别为 80%和 90%时,其能源利用率最高.其他 1/3 的服务器的最优点为(0.65,0.7),剩余 1/3 的服务最优点为(0.5,0.5).

服务器的初始硬盘利用率是[0,0.3]范围内的随机数,初始 CPU 利用率为[0,0.35]范围内的随机数.为了更好地体现算法的有效性,我们设置了一组特殊的服务器,它们的 CPU 利用率分别如下:$CS[5]=0.5$,$CS[25]=0.7$,$CS[45]=0.9$,$CS[75]=0.5$,$CS[95]=0.7$,$CS[115]=0.9$,$CS[145]=0.5$,$CS[165]=0.7$,$CS[195]=0.9$;同样地,我们设置了另一组特殊的服务器,它们的硬盘利用率分别如下:$HS[10]=0.5$,$HS[30]=0.65$,$HS[50]=0.8$,$HS[80]=0.5$,$HS[100]=0.65$,$HS[120]=0.8$,$HS[150]=0.5$,$HS[170]=0.65$,$HS[190]=0.8$.我们将本文提出的算法 MCMC 与 Pμ-CHC 调度算法进行了 3 组对比实验,每组实验每个算法独立运行 30 次,以获取并记录实验结果.

- 首先,本文提出的算法比 Pμ-CHC 在提高服务器的能源利用率方面更加有效,因为本节提出的算法,其能源无效值低至 0.496 3,而 Pμ-CHC 获得的能源无效值高达 13.638 3;
- 其次,MCMC 算法可以达到任务 10%数据本地化执行率,而 Pμ-CHC 则不能.
- 再次,将服务器划分为 3 组,3 组服务器的最优 CPU 利用率分别是 0.9,0.7 和 0.5,最优硬盘利用率分别是 0.8,0.65 和 0.5.实验数据表明,明显优于 Pμ-CHC.MCMC 调度后,3 组服务器的 CPU 利用率分别位于 0.9,0.7 和 0.5 左右,都接近于其最优值;同时,硬盘利用率分别位于 0.8,0.65 和 0.5 左右,也都接近其最优值.这是因为针对适量负载,Pμ-CHC 并没有考虑服务器的资源利用率:它为每台服务器只是按照负载

均衡的方式分配数据和任务.

4.3 可扩展性

实验首先依从文献[25],将调度算法运行的总体时间阈值定为 90s;然后,针对每维的一致性、半一致性和不一致性实例,采用不同的子群个数([2,16])进行实验,直到调度算法运行到时间阈值.这里,任务调度时间的标准化是采用 n 个子群实现时间与一个子群实现时间的比率.图 5 显示出相关实验结果.

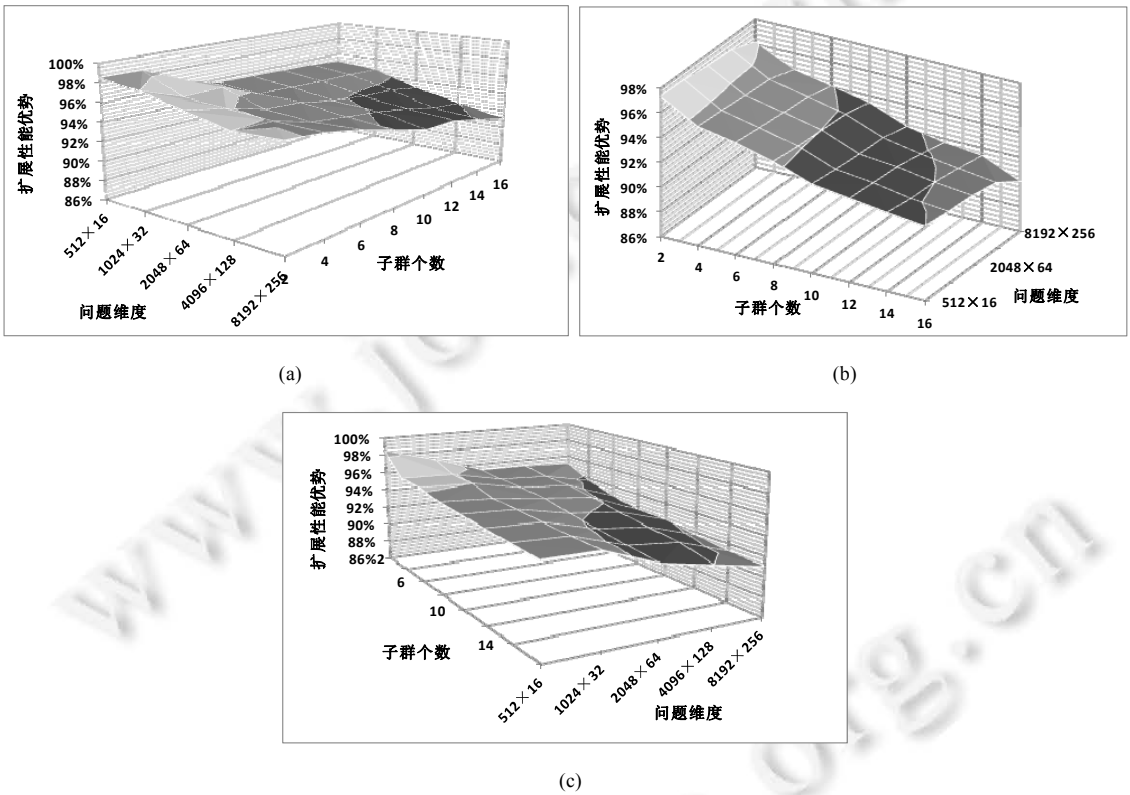

Fig.5 Improvements of MCMC over the algorithms summarized for scalability analysis and parallel performance

图 5 不同算法可扩展及并行方面性能比较

图 5 表明:随着云调度问题维数的增加,算法 MCMC 求得的任务调度时间明显下降,时间性能优势明显.这主要是因为 MCMC 包含种群和智能的双重协同进化过程,可实现种群加速收敛及个体多样性保持.

5 结 论

本文对融合能效感知的云调度进行研究,有效降低数据密集应用的通信开销、兼顾提供者和消费者双方的利益,并保证系统双层负载均衡性.随着新应用(数据密集应用、计算密集应用)、新环境(计算网格、云计算)和新性能指标(能效)的出现,分布式计算资源管理核心技术之异构调度研究具有重大的理论和应用价值.

References:

[1] Zhong L, Luo Q, Wen D, Qiao SD, Shi JM, Zhang WM. A task assignment algorithm for multiple aerial vehicles to attack targets with dynamic values. IEEE Trans. on Intelligent Transportation Systems, 2013,14(1):236−248. [doi; 10.1109/TITS.2012.2210882]

[2] Huang ZX, Lu XD, Duan HL. A task operation model for resource allocation optimization in business process management. IEEE Trans. on Systems, Man and Cybernetics, Part A: Systems and Humans, 2012,42(5):1256−1270. [doi; 10.1109/TSMCA.2012.2187889]

[3] Zhu H, Wang YP. Integration of security grid dependent tasks scheduling double-objective optimization model and algorithm. Ruan Jian Xue Bao/Journal of Software, 2011,22(11): 2729−2748 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/3900.htm [doi: 10.3724/S.P.J.1001.2011.03900]

[4] Tang J, Lim MH, Ong YS. Diversity-Adaptive parallel memetic algorithm for solving large scale combinatorial optimization problems. Soft Computing, 2007,11(9):873−888. [doi; 10.1007/s00500-006-0139-6]

[5] Dou H, Ji Y, Wang PJ, Zhang KY. Workload scheduling algorithm for minimizing electricity bills of green data centers. Ruan Jian Xue Bao/Journal of Software, 2014,25(7):1448−1458 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4599.htm [doi: 10.13328/j.cnki.jos.004599]

[6] Braun TD, Siegel HJ, Beck N. A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems. Journal of Parallel and Distributed Computing, 2001,61(6):810−837. [doi; 10.1006/jpdc.2000.1714]

[7] Hsieh FS, Lin JB. A dynamic scheme for scheduling complex tasks in manufacturing systems based on collaboration of agents. Applied Intelligence, 2014,41(2):366−382. [doi; 10.1007/s10489-014-0521-5]

[8] Guérout T, Medjiah S, Costa GD, Monteil T. Quality of service modeling for green scheduling in clouds. Sustainable Computing: Informatics and Systems, 2014,4225−240. [doi; 10.1016/j.suscom.2014.08.006]

[9] 张冬松.多核多处理器系统的节能实时调度技术研究[博士学位论文].长沙:国防科学技术大学,2012.

[10] IBM Blue Gene Team. Overview of the IBM Blue Gene/P project. IBM Journal of Research and Development, 2008,52(1):199−220.

[11] Li M, Liu BJ, Yao FF. Min-energy voltage allocation for tree structured tasks. Journal of Combinatorial Optimization, 2006, 11(3):305−319. [doi; 10.1007/s10878-006-7910-6]

[12] Li M, Yao FF. An efficient algorithm for computing optimal discrete voltage schedules. SIAM Journal on Computing, 2005,35(3):658−671. [doi; 10.1137/050629434]

[13] Irani S, Shukla S, Gupta R. Algorithms for power savings. ACM Trans. on Algorithms, 2007,3(4):1−23. [doi; 10.1145/1290672.1290678]

[14] Han X, Lam TW, Lee LK, Isaac KK, Prudence T, Wong WH. Deadline scheduling and power management for speed bounded processors. Theor. Comput. Sci., 2010,411(40-42):3587−3600. [doi; 10.1016/j.tcs.2010.05.035]

[15] Goncalves JF, de Magalhaes Mendes JJ, Resende MGC. A hybrid genetic algorithm for the job shop scheduling problem. European Journal of Operational Research, 2005,167(1):77−95. [doi; 10.1016/j.ejor.2004.03.012]

[16] Lin JY, Chen YP. Analysis on the collaboration between global search and local search in memetic computation. IEEE Trans. on Evolutionary Computation, 2011,15(5):608−623. [doi; 10.1109/TEVC.2011.2150754]

[17] Li B, Zhou Z, Zou WX, Li DJ. Quantum memetic evolutionary algorithm-based low-complexity signal detection for underwater acoustic sensor networks. IEEE Trans. on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 2012,42(5):626−640. [doi; 10.1109/TSMCC.2011.2176486]

[18] Wen Y, Xu H, Yang JD. A heuristic-based hybrid genetic-variable neighborhood search algorithm for task scheduling in heterogeneous multiprocessor system. Information Sciences, 2011,181(3):567−581. [doi; 10.1016/j.ins.2010.10.001]

[19] Sancho SS, Xu Y, Yao X. Hybrid meta-heuristics algorithms for task assignment in heterogeneous computing systems. Computers & Operations Research, 2006,33(3):820−835. [doi; 10.1016/j.cor.2004.08.010]

[20] Chitra P, Rajaram R, Venkatesh P. Application and comparison of hybrid evolutionary multiobjective optimization algorithms for solving task scheduling problem on heterogeneous systems. Applied Soft Computing, 2011,11(2):2725−2734. [doi; 10.1016/j.asoc.2010.11.003]

[21] Gong YJ, Zhang J, Chung SH, Chen WN, Zhan ZH, Li Y, Shi YH. An efficient resource allocation scheme using particle swarm optimization. IEEE Trans. on Evolutionary Computation, 2012,16(6):801−816. [doi; 10.1109/TEVC.2012.2185052]

[22] Wu AS, Yu H, Jin SY, Lin KC, Schiavone G. An incremental genetic algorithm approach to multiprocessor scheduling. IEEE Trans. on Parallel and Distributed Systems, 2004,15(9):824−834. [doi; 10.1109/TPDS.2004.38]

[23] Braun TD, Siegel HJ, Maciejewski AA, Hong Y. Static resource allocation for heterogeneous computing environments with tasks having dependencies, priorities, deadlines, and multiple versions. Journal of Parallel and Distributed Computing, 2008,68(11): 1504−1516. [doi; 10.1016/j.jpdc.2008.06.006]

[24] De Falco I, Della Cioppa A, Maisto D, Scafuri U, Tarantino E. Biological invasion-inspired migration in distributed evolutionary algorithms. Information Sciences, 2012,207(10):50−65. [doi; 10.1016/j.ins.2012.04.027]

[25] Nesmachnow S, Cancela H, Alba E. A parallel micro evolutionary algorithm for heterogeneous computing and grid scheduling. Applied Soft Computing, 2012,12(2):626−639. [doi; 10.1016/j.asoc.2011.09.022]

[26] Estahbanati MJ. Hybrid probabilistic-harmony search algorithm methodology in generation scheduling problem. Journal of Experimental & Theoretical Artificial Intelligence, 2014,26(2):283−296. [doi; 10.1080/0952813X.2013.861876]

[27] Pant A, Jafri H. Communicating efficiently on cluster based grids with MPICH-VMI. In: Proc. of the IEEE Int'l Conf. on Cluster Computing. San Diego: IEEE, 2004. 23−33. [doi; 10.1109/CLUSTR.2004.1392598]

[28] Ali S, Siegel HJ, Maheswaran M, Hensgen D, Ali S. Task execution time modeling for heterogeneous computing systems. In: Proc. of the 9th Heterogeneous Computing Workshop. Washington: IEEE, 2000. 185−199. [doi; 10.1109/HCW.2000.843743]

附中文参考文献:

[3] 朱海,王宇平.融合安全的网格依赖任务调度双目标优化模型及算法.软件学报,2011,22(11):2729−2748. http://www.jos.org.cn/1000-9825/3900.htm [doi: 10.3724/S.P.J.1001.2011.03900]

[5] 窦晖,齐勇,王培健,张恺玉.一种最小化绿色数据中心电费的负载调度算法.软件学报,2014,25(7):1448−1458. http://www.jos.org.cn/1000-9825/4599.htm [doi: 10.13328/j.cnki.jos.004599]

王静莲(1979－),女,山东莱州人,博士,讲师,主要研究领域为并行与高性能计算,绿色计算.

刘弘(1955－),女,博士,教授,博士生导师,主要研究领域为人工智能,多目标全局优化算法.

龚斌(1964－),男,博士,教授,博士生导师,主要研究领域为高性能计算,机群计算.

李少辉(1978－),男,讲师,主要研究领域为绿色计算,多目标全局优化算法.